Slawomir Koziel
Leifur Leifsson *Editors*

Surrogate-Based Modeling and Optimization

Applications in Engineering

Springer

Surrogate-Based Modeling and Optimization

Slawomir Koziel • Leifur Leifsson
Editors

Surrogate-Based Modeling and Optimization

Applications in Engineering

Springer

Editors
Slawomir Koziel
Engineering Optimization & Modeling Cent
Reykjavik University
Reykjavik, Iceland

Leifur Leifsson
School of Science and Engineering,
Engineering Optimization & Modeling
Reykjavik University
Reykjavik, Iceland

ISBN 978-1-4614-7550-7 ISBN 978-1-4614-7551-4 (eBook)
DOI 10.1007/978-1-4614-7551-4
Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2013941933

Mathematics Subject Classification (2010): 74P10, 80M50, 97M10, 62P30

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Contemporary engineering design is heavily based on computer simulations. Accurate, high-fidelity simulations are used not only for design verification but, most importantly, to adjust parameters of the system (e.g., wing geometry, material parameters of antennas) so that it meets given performance requirements. Unfortunately, accurate simulations are often computationally expensive, with evaluation times ranging from hours to days per design. Consequently, design automation using conventional optimization methods is often impractical or even prohibitive. Other issues include the numerical noise that is often present in simulation responses, and the absence of sensitivity information. These, and other problems, can be alleviated by the development and employment of so-called surrogates, which reliably represent the expensive, simulation-based model of the system/device of interest, but are much cheaper and analytically tractable.

This edited book is about surrogate-based modeling and optimization techniques and their applications for solving difficult and computationally expensive engineering design problems. A group of international experts summarize recent developments in the field and demonstrate applications in various disciplines of engineering and science. The main purpose of the work is to provide the basic concepts and formulations of the surrogate-based modeling and optimization paradigm, as well as to discuss relevant modeling techniques, optimization algorithms, and design procedures.

Simulation-driven design based on surrogate models plays an increasingly important role in contemporary engineering and permits us to solve problems that cannot be solved otherwise, particularly by dramatically reducing the computational cost of the solution process. Unfortunately, recent results concerning surrogate-based modeling and optimization are scattered throughout the literature in various engineering fields and, therefore, are not easily accessible to a reader interested in this technology. Thus, this book should be of interest to engineers from any discipline where computationally heavy simulations (such as finite element, computational fluid dynamics, and computational electromagnetics analyses) are used on a daily basis in the design process. The editors of this volume hope that the presented material will allow the readers to gain an understanding of the basic mechanisms of the surro-

gate modeling process and familiarize themselves with important components of surrogate-based optimization algorithms and the advantages of employing variable-fidelity simulation-driven design, as well as enable them to reduce the cost of the design process aided by computer simulations.

Reykjavik, Iceland
March 2013

Slawomir Koziel
Leifur Leifsson

Contents

Space Mapping for Electromagnetic-Simulation-Driven Design Optimization

Slawomir Koziel, Leifur Leifsson, and Stanislav Ogurtsov

Abstract Space mapping (SM) has been one of the most popular surrogate-based optimization techniques in microwave engineering to date. By exploiting the knowledge embedded in the underlying coarse model (e.g., an equivalent circuit), SM allows dramatic reduction of the computational cost while optimizing electromagnetic (EM)-simulated structures such as filters or antennas. While potentially very efficient, SM is not always straightforward to implement and set up, and may suffer from convergence problems. In this chapter, we discuss several variations of an SM optimization algorithm aimed at improving SM performance for design problems involving EM simulations. These include SM with constrained parameter extraction and surrogate model optimization designed to overcome the problem of selecting preassigned parameters for implicit SM, SM with response surface approximation coarse models that maintain SM efficiency when a fast coarse model is not available, and SM with sensitivity which takes advantage of adjoint sensitivity (which has recently become commercially available in EM simulators) to improve the convergence properties and further reduce the computational cost of SM algorithms. Each variation of the SM algorithm presented here is illustrated using a real-world microwave design example.

Keywords Computer-aided design (CAD) · Microwave engineering · Simulation-driven optimization · Electromagnetic (EM) simulation · Surrogate-based optimization · Space mapping · Surrogate model · High-fidelity model · Coarse model

1 Introduction

Space mapping (SM) [1–3] was originally developed in the 1990s to deal with computationally expensive design problems in microwave engineering [3, 4]. Automated design closure of structures evaluated using electromagnetic (EM) simu-

S. Koziel (✉) · L. Leifsson · S. Ogurtsov
Engineering Optimization & Modeling Center, School of Science and Engineering, Reykjavik University, Menntavegur 1, 101 Reykjavik, Iceland
e-mail: koziel@ru.is

S. Koziel, L. Leifsson (eds.), *Surrogate-Based Modeling and Optimization*,
DOI 10.1007/978-1-4614-7551-4_1,

lations is still a challenging task today, mostly due to the high computational cost of accurate, high-fidelity EM simulation. The presence of massive computing resources does not always translate into computational speedup. This is due to a growing demand for simulation accuracy (which requires, among other things, finer discretization of the structure), as well as the necessity of including important interactions between the structure under design and its environment (e.g., antenna housing and connectors, etc.), which increases the computational domain and, consequently, slows down the simulation. At the same time, multiphysics simulations (e.g., including thermal effects) become more and more important, further contributing to the computational cost of the simulation. As conventional optimization algorithms (e.g., gradient-based schemes with numerical derivatives) require tens, hundreds, or even thousands of objective function calls per run (depending on the number of design variables), the computational cost of the whole optimization process may not be acceptable.

SM belongs to a broader family of methods called surrogate-based optimization (SBO) techniques [5–7]. SM and most other SBO methods share a main structure, in which the direct optimization of the expensive (here, EM-simulated) structure, referred to as the high-fidelity or fine model, is replaced by iterative refinement and reoptimization of a low-fidelity (or coarse) model. The coarse model is a physics-based representation of the high-fidelity one. It is less accurate but supposedly much faster than the latter. An example of a coarse model in microwave engineering is an equivalent circuit which describes the same structure as the fine model but using circuit theory rather than full-wave EM simulation. The SM surrogate is constructed by enhancing the coarse model through auxiliary transformations, usually linear, with parameters of these transformations obtained in what is called the parameter extraction (PE) process [8], which is a trademark of SM. PE is executed to reduce the misalignment between the responses of the space-mapped coarse model and the fine model at a limited number of designs, usually those that have already emerged during the SM optimization run. The benefit of SM lies in the fact that each SM iteration usually requires evaluation the high-fidelity model at a single design (the one obtained by optimizing the current surrogate model), and—for a well-performing algorithm—only a few iterations are necessary to yield a satisfactory design.

SM has been successfully applied to optimize a number of microwave components, the majority of which are filters [1–4] and impedance transformers [8], but also antennas [9–11], etc. As mentioned before, one of the fundamental prerequisites of SM is that the underlying coarse model be fast, so that the computational overhead related to parameter extraction and surrogate model optimization performed at each iteration of the algorithm can be neglected. For this reason, equivalent circuit models are preferred. Unfortunately, reliable equivalent circuit models are not available for many structures, including broadband antennas [12] and substrate integrated structures [13]. Also, if EM coupling between the device of interest and its environment has to be taken into account (e.g., an antenna mounted in a cellphone or on a vehicle), full-wave simulation is probably the only way to evaluate the system. A possible lack of fast models poses a difficulty for SM, because the computational overhead related to multiple evaluation of the low-fidelity model, particularly due to PE, may determine the total CPU cost of the SM process.

Another issue is convergence: as SM does not use sensitivity information by default, it is not globally convergent in a conventional sense, unless certain conditions regarding similarity between the low- and high-fidelity models [14] are met. Those conditions are, unfortunately, difficult to verify in practice [15]. At the same time, SM offers a number of ways of constructing the surrogate model. A number of elementary transformations (input SM [8], implicit SM [16], or frequency SM [2, 14]) can be combined into more involved models. The specific choice of the model affects the algorithm performance [17], and the optimal choice is not trivial [15].

In this chapter, we discuss several variations of the SM algorithms specifically targeted at improving the SM performance for microwave engineering applications. The chapter is organized as follows. In Sect. 2, we recall the SM concept and formulate a generic SM algorithm. In Sect. 3, we describe implicit SM with constrained parameter extraction, which helps improve convergence of the SM algorithm without requiring careful selection of its preassigned parameters. Section 4 is devoted to SM with auxiliary response surface approximation coarse models. This approach aims at improving the efficiency of the SM optimization process in situations when the available low-fidelity models are relatively expensive (e.g., obtained through coarse-discretization EM simulations), which is usually the case in antenna design. In Sect. 5, we discuss the enhancement of SM using adjoint sensitivity. By exploiting derivative information whenever possible, specifically, to enhance the surrogate model as well as to speed up both PE and surrogate model optimization, it is possible to improve the algorithm convergence and to reduce its computational cost. Section 6 concludes the chapter with some recommendations for readers interested in applying SM in their design work.

2 Space Mapping Optimization: A Brief Introduction

In this section, we briefly recall the formulation of the generic space mapping (SM) algorithm, the concept of parameter extraction (PE), and the basic types of SM surrogate models.

A microwave design task can be formulated as a nonlinear minimization problem

$$\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x} \in X_f} U\left(\boldsymbol{R}_f(\boldsymbol{x})\right), \tag{1}$$

where $\boldsymbol{R}_f \in R^m$ denotes the response vector of the device of interest, e.g., the modulus of the transmission coefficient $|S_{21}|$ evaluated at m different frequencies. U is a given scalar merit function, e.g., a minimax function with upper and lower specifications [8]. Vector $\boldsymbol{x}^*$ is the optimal design to be determined. Normally, $\boldsymbol{R}_f$ is obtained through computationally expensive electromagnetic (EM) simulation, and is referred to as the high-fidelity or fine model. Because of the high computational cost, using conventional optimization techniques to handle (1) may be impractical; both gradient-based (e.g., quasi-Newton [18]) and derivative-free (pattern search [19], genetic algorithms [20]) methods usually require a substantial number of objective function (and thus, high-fidelity model) evaluations.

SM [1–3, 21] is a methodology that aims at reducing the cost of solving the problem (1). SM belongs to a broader class of surrogate-based optimization (SBO) techniques [7–9]. Similarly to other SBO methods, SM speeds up the design process by shifting the optimization burden to an inexpensive yet reasonably accurate surrogate model of the device of interest. In the generic SM framework described here the direct optimization of the computationally expensive EM-simulated high-fidelity model $\boldsymbol{R}_f$ is replaced by an iterative procedure [7, 14]

$$\boldsymbol{x}^{(i+1)} = \arg\min_{\boldsymbol{x}} U\left(\boldsymbol{R}_s^{(i)}(\boldsymbol{x})\right), \tag{2}$$

that generates a sequence of points (designs) $\boldsymbol{x}^{(i)} \in X_f$, $i = 0, 1, \ldots$, that are approximate solutions to the original design problem (1). Each $\boldsymbol{x}^{(i+1)}$ is the optimal design of the surrogate model $\boldsymbol{R}_s^{(i)} : X_s^{(i)} \to R^m$, $X_s^{(i)} \subseteq R^n$, $i = 0, 1, \ldots$. $\boldsymbol{R}_s^{(i)}$ is assumed to be a computationally cheap and sufficiently reliable representation of the fine model $\boldsymbol{R}_f$, particularly in the neighborhood of the current design $\boldsymbol{x}^{(i)}$. Under these assumptions, the algorithm (2) is likely to produce a sequence of designs that quickly approach $\boldsymbol{x}_f^*$.

Typically, $\boldsymbol{R}_f$ is only evaluated once per iteration (at every new design $\boldsymbol{x}^{(i+1)}$) for verification purposes and to obtain the data necessary to update the surrogate model. Since the surrogate model is computationally cheap, its optimization cost (cf. (2)) can usually be neglected, and the total optimization cost is determined by the evaluation of $\boldsymbol{R}_f$. The key point here is that the number of evaluations of $\boldsymbol{R}_f$ for a well-performing surrogate-based algorithm is substantially smaller than for any direct optimization method (e.g., a gradient-based one). Figure 1 shows the block diagram of the SM optimization process.

If the surrogate model satisfies zero- and first-order consistency conditions with the fine model, i.e., $\boldsymbol{R}_s^{(i)}(\boldsymbol{x}^{(i)}) = \boldsymbol{R}_f(\boldsymbol{x}^{(i)})$ and $(\partial \boldsymbol{R}_s^{(i)}/\partial \boldsymbol{x})(\boldsymbol{x}^{(i)}) = (\partial \boldsymbol{R}_f/\partial \boldsymbol{x})(\boldsymbol{x}^{(i)})$ (verification of the latter requires $\boldsymbol{R}_f$ sensitivity data), and the algorithm (2) is enhanced by the trust region method [22], then it is provably convergent to a local fine model optimum [23]. Convergence can also be guaranteed if the algorithm (2) is enhanced by properly selected local search methods [24]. SM [8, 14, 25, 26] does not normally rely on the aforementioned enhancements; however, it requires the surrogate model to be constructed from the physically based coarse model [8]. This usually gives remarkably good performance in the sense of the SM algorithm being able to quickly locate a satisfactory design.

The way that SM constructs the surrogate model for the iterative process (2) is one of the features that distinguish this technique from many other SBO approaches. The surrogate model at iteration i, $\boldsymbol{R}_s^{(i)}$, is constructed from the low-fidelity model so that the misalignment between $\boldsymbol{R}_s^{(i)}$ and the fine model is minimized using the parameter extraction (PE) process, which is the nonlinear minimization problem by itself [8]. The surrogate is defined as [14]

$$\boldsymbol{R}_s^{(i)}(\boldsymbol{x}) = \boldsymbol{R}_{s.g}\left(\boldsymbol{x}, \boldsymbol{p}^{(i)}\right), \tag{3}$$

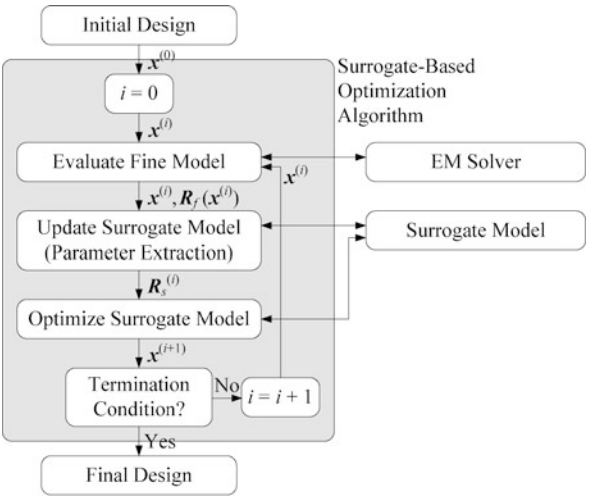


Fig. 1 Surrogate-based simulation-driven design optimization: the optimization burden is shifted to the computationally cheap surrogate model which is updated and reoptimized at each iteration of the main optimization loop. High-fidelity EM simulation is only performed once per iteration to verify the design produced by the surrogate model and to update the surrogate itself. The number of iterations for a well-performing SBO algorithm is substantially smaller than for conventional techniques

where $\boldsymbol{R}_{s.g}$ is a generic SM surrogate model, i.e., the low-fidelity model composed with suitable transformations, whereas

$$\boldsymbol{p}^{(i)} = \arg\min_{\boldsymbol{p}} \sum_{k=0}^{i} w_{i.k} \left\| \boldsymbol{R}_f\left(\boldsymbol{x}^{(k)}\right) - \boldsymbol{R}_{s.g}\left(\boldsymbol{x}^{(k)}, \boldsymbol{p}\right) \right\| \tag{4}$$

is a vector of model parameters and $w_{i.k}$ are weighting factors; a common choice of $w_{i.k}$ is $w_{i.k} = 1$ for all i and all k.

Various SM surrogate models are available [8, 14]. They can be roughly categorized into four groups:

- Models based on a (usually linear) distortion of coarse model parameter space, e.g., input SM of the form $\boldsymbol{R}_{s.g}(\boldsymbol{x}, \boldsymbol{p}) = \boldsymbol{R}_{s.g}(\boldsymbol{x}, \boldsymbol{B}, \boldsymbol{c}) = \boldsymbol{R}_c(\boldsymbol{B} \cdot \boldsymbol{x} + \boldsymbol{c})$ [8];
- Models based on a distortion of the coarse model response, e.g., output SM of the form $\boldsymbol{R}_{s.g}(\boldsymbol{x}, \boldsymbol{p}) = \boldsymbol{R}_{s.g}(\boldsymbol{x}, \boldsymbol{d}) = \boldsymbol{R}_c(\boldsymbol{x}) + \boldsymbol{d}$ [14];
- Implicit SM, where the parameters used to align the surrogate with the fine model are separate from the design variables, i.e., $\boldsymbol{R}_{s.g}(\boldsymbol{x}, \boldsymbol{p}) = \boldsymbol{R}_{s.g}(\boldsymbol{x}, \boldsymbol{x}_p) = \boldsymbol{R}_{c.i}(\boldsymbol{x}, \boldsymbol{x}_p)$, with $\boldsymbol{R}_{c.i}$ being the coarse model dependent on both the design variables $\boldsymbol{x}$ and the preassigned parameters $\boldsymbol{x}_p$ (e.g., dielectric constant, substrate height) that are normally fixed in the fine model but can be freely altered in the coarse model [14];
- Custom models exploiting parameters characteristic to a given design problem; the most characteristic example is frequency SM $\boldsymbol{R}_{s.g}(\boldsymbol{x}, \boldsymbol{p}) = \boldsymbol{R}_{s.g}(\boldsymbol{x}, \boldsymbol{F}) =$

$\boldsymbol{R}_{c.f}(\boldsymbol{x}, \boldsymbol{F})$ [8], where $\boldsymbol{R}_{c.f}$ is a frequency-mapped coarse model, i.e., the coarse model evaluated at frequencies ω different from the original frequency sweep for the fine model, according to the mapping $\omega \rightarrow f_1 + f_2\omega$, with $\boldsymbol{F} = [f_1\, f_2]^T$.

SM usually comprises combined transformations. For instance, a surrogate model employing input, output, and frequency SM transformations would be $\boldsymbol{R}_{s.g}(\boldsymbol{x}, \boldsymbol{p}) = \boldsymbol{R}_{s.g}(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{F}) = \boldsymbol{R}_{c.f}(\boldsymbol{x} + \boldsymbol{c}, \boldsymbol{F}) + \boldsymbol{d}$. The rationale for this is that a properly chosen mapping may significantly improve the performance of the SM algorithm; however, the optimal selection of the mapping type for a given design problem is not trivial [15]. Work has been done to ease the selection process for a given design problem [17, 27]. However, regardless of the mapping choice, coarse model accuracy is what principally affects the performance of the SM design process. One can quantify the quality of the surrogate model through rigorous convergence conditions [15]. These conditions, although useful for developing more efficient SM algorithms and automatic surrogate model selection techniques, cannot usually be verified because of the limited amount of data available from the fine model. In practice, the most important criterion for assessing the quality or accuracy of the coarse model is still visual inspection of the fine and coarse model responses at certain points and/or examining absolute error measures such as $\|\boldsymbol{R}_f(\boldsymbol{x}) - \boldsymbol{R}_c(\boldsymbol{x})\|$.

The coarse model is the most important factor that affects the performance of the SM algorithm. The first stems from accuracy. Coarse model accuracy (more generally, the accuracy of the SM surrogate [15]) is the main factor that determines the efficiency of the algorithm in terms of finding a satisfactory design. The more accurate the coarse model, the smaller the number of fine model evaluations necessary to complete the optimization process. If the coarse model is insufficiently accurate, the SM algorithm may need more fine model evaluations or may even fail to find a good quality design.

The second important characteristic is the evaluation cost. It is essential that the coarse model be computationally much cheaper than the fine model, because both PE (4) and surrogate optimization (2) require large numbers of coarse model evaluations. Ideally, the evaluation cost of the coarse model should be negligible when compared to the evaluation cost of the fine model, in which case the total computational cost of the SM optimization process is merely determined by the necessary number of fine model evaluations. If the evaluation time of the coarse model is too high, say, larger than 1 % of the fine model evaluation time, the computational cost of surrogate model optimization and, especially, PE, start playing important roles in the total cost of SM optimization and may even determine it. Therefore, practical applicability of SM is limited to situations where the coarse model is computationally much cheaper than the fine model. The majority of SM models reported in the literature (e.g., [4, 8, 14]) concern microstrip filters, transformers, or junctions where fast and reliable equivalent circuit coarse models are easily available.

3 Implicit Space Mapping with Constrained Parameter Extraction for Microwave Filter Design

In this section, we discuss SM with constrained PE [28] as a way to alleviate the problems related to the proper selection of the SM parameters used to construct the surrogate model as well as to improve convergence properties of the SM algorithm. Without loss of generality, we restrict our considerations to implicit SM [16], the most general approach to SM. Implicit SM allows us to introduce any number of surrogate model parameters. In particular, over-flexibility of the surrogate model required by our method can be easily obtained. Also, implicit SM can incorporate most other SM types, including input and output SM.

3.1 Implicit Space Mapping Algorithm

Implicit SM follows the generic SM scheme (2); however, for better clarity, we assume here a specific form of the SM surrogate model. Let $\boldsymbol{R}_c : X_c \times X_p \to R^m, \in R^m$ denote the response vector of the coarse model that describes the same object as the fine model: less accurate but much faster to evaluate. $\boldsymbol{R}_c$ depends on two sets of parameters: (i) design variables $\boldsymbol{x}$, the same as in the fine model, and (ii) preassigned parameters $\boldsymbol{x}_p$, i.e., parameters that are normally fixed in the fine model (e.g., dielectric constants) but can be adjusted in the coarse model to match it to $\boldsymbol{R}_f$.

The generic SM algorithm (2) can then be reformulated as follows:

$$\boldsymbol{x}^{(i+1)} = \arg\min_{\boldsymbol{x}} U\left(\boldsymbol{R}_c\left(\boldsymbol{x}, \boldsymbol{x}_p^{(i)}\right)\right), \tag{5}$$

where

$$\boldsymbol{x}_p^{(i)} = \arg\min_{\boldsymbol{x}_p} \left\| \boldsymbol{R}_f\left(\boldsymbol{x}^{(i)}\right) - \boldsymbol{R}_c\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}_p\right) \right\|. \tag{6}$$

Thus, the coarse model with adjusted values of preassigned parameters becomes a surrogate model for the implicit SM algorithm. We note that other types of SM can also be reformulated in terms of implicit SM. For example, the surrogate model of the most popular input SM [14] is defined as $\boldsymbol{R}_s(\boldsymbol{x}) = \boldsymbol{R}_c(\boldsymbol{x} + \boldsymbol{c})$, where $\boldsymbol{c}$ is an $n \times 1$ vector of parameters. It can be formulated in implicit SM terms as $\boldsymbol{R}_{c.i}(\boldsymbol{x}, \boldsymbol{x}_p)$, where $\boldsymbol{R}_{c.i}$ is the implicit-SM-like coarse model defined as $\boldsymbol{R}_{c.i}(\boldsymbol{x}, \boldsymbol{x}_p) = \boldsymbol{R}_c(\boldsymbol{x} + \boldsymbol{x}_p)$. In particular, the same software implementation can be used to realize our algorithm for both implicit and input cases. Analogous reformulations can be defined for scaling-like input SM $\boldsymbol{R}_s(\boldsymbol{x}) = \boldsymbol{R}_c(\boldsymbol{B} \cdot \boldsymbol{x})$, where $\boldsymbol{B}$ is an $n \times n$ matrix of SM parameters, output [14] and frequency SM [14].

3.2 Space Mapping with Constrained Parameter Extraction

The performance of SM algorithms depends heavily on the quality of the coarse model utilized in the optimization process and the type of mapping involved in

creating the surrogate model [15]. For implicit SM, the number of possible ways of selecting preassigned parameters is virtually unlimited. However, it is not obvious how to select a set of parameters that might allow the surrogate to closely approximate the fine model and simultaneously have good generalization capability. A wrong choice of the parameter set may result in inadequate performance of the SM algorithm, including convergence issues, poor quality of the final design, and excessive computational cost [17].

Various ways of alleviating this problem have been proposed. They include adaptive SM algorithms [17] and assessment methodologies [15, 29]. Although useful in the selection of the surrogate model and its parameters, none of these techniques guarantees algorithm convergence and overall good performance. On the other hand, trust region enhanced SM algorithms ensure algorithm convergence but not always a sufficient quality of the final design, because in the case of inadequate improvement of the objective function [30], they force the algorithm to terminate. Also, the trust region approach increases the computational cost of the SM optimization process.

The technique described in this section controls the trade-off between the approximation and generalization capability of the surrogate model by properly restricting the parameter space of the model. Theoretical justification of the method can be found in [28].

It is assumed that the initial surrogate model, i.e., the coarse model without any constraints on its preassigned parameters, is able to approximate the fine model with sufficient accuracy. This accuracy can be measured at a given iteration point $\boldsymbol{x}^{(i)}$, using any suitable criteria, e.g., $\varepsilon^{(i)} = \|\boldsymbol{R}_f(\boldsymbol{x}^{(i)}) - \boldsymbol{R}_c(\boldsymbol{x}^{(i)}, \boldsymbol{x}_p^{(i)})\|_p$, where $\|\cdot\|_p$ determines the norm type (e.g., $\|\cdot\|_2$ for the Euclidean norm, or $\|\cdot\|_\infty$ for the maximum norm). We would like $\varepsilon^{(i)}$ to be small, $\varepsilon^{(i)} \leq \varepsilon_{\max}$, where $\varepsilon_{\max}$ is a user-defined threshold value, so that the surrogate model is a sufficiently good representation of $\boldsymbol{R}_f$.

It is normally feasible to build a surrogate model satisfying the above requirements. Because $\boldsymbol{R}_c$ is physically based, its response is similar to that of $\boldsymbol{R}_f$. An appropriate surrogate model can then be created by introducing a sufficient number of preassigned parameters, e.g., dielectric constants and substrate heights corresponding, if necessary, to individual components of the microwave structure in question. Even synthetic parameters can be used (i.e., parameters not corresponding to any physical parameter in $\boldsymbol{R}_f$ but used to increase model flexibility, e.g., a small capacitor introduced between coupled lines in the microstrip filter model). A rule of thumb is that the number of parameters should be larger than the number of design variables. If necessary, other SM transformations (e.g., input, frequency) can be incorporated (cf. Sect. 2).

According to the algorithm proposed in [31], the parameter extraction (PE) process (6) is replaced by the following constrained version:

$$\boldsymbol{x}_p^{(i)} = \arg \min_{\boldsymbol{l}^{(i)} \leq \boldsymbol{x}_p \leq \boldsymbol{u}^{(i)}} \left\| \boldsymbol{R}_f\left(\boldsymbol{x}^{(i)}\right) - \boldsymbol{R}_c\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}_p\right) \right\|, \tag{7}$$

where $\boldsymbol{l}^{(i)}$ and $\boldsymbol{u}^{(i)}$ are lower and upper bounds for the preassigned parameters at iteration i. We assume here that $\boldsymbol{l}^{(i)} = \boldsymbol{x}_p^{(i-1)} - \boldsymbol{\delta}^{(i)}$ and $\boldsymbol{u}^{(i)} = \boldsymbol{x}_p^{(i-1)} + \boldsymbol{\delta}^{(i)}$, where

$\boldsymbol{x}_p^{(i-1)}$ is the vector of model parameters at iteration $i-1$ ($\boldsymbol{x}_p^{(0)}$ represents the initial values of the preassigned parameters), whereas $\boldsymbol{\delta}^{(i)}$ is a vector representing the parameter space size ($\boldsymbol{\delta}^{(0)}$ is a user-defined initial value).

At iteration i, the algorithm adjusts $\boldsymbol{l}^{(i)}$ and $\boldsymbol{u}^{(i)}$ and performs PE as follows ($\boldsymbol{\delta}^{(i)}$, $\boldsymbol{x}_p^{(i-1)}$ and $\varepsilon_{\max}$ are input arguments):

1. Calculate $\boldsymbol{l}^{(i)} = \boldsymbol{x}_p^{(i-1)} - \boldsymbol{\delta}^{(i)}$ and $\boldsymbol{u}^{(i)} = \boldsymbol{x}_p^{(i-1)} + \boldsymbol{\delta}^{(i)}$;
2. Find $\boldsymbol{x}_p^{(i)}$ using (7);
3. If $\varepsilon^{(i)} \le \alpha_{\text{decr}} \cdot \varepsilon_{\max}$ then $\boldsymbol{\delta}^{(i+1)} = \boldsymbol{\delta}^{(i)} / \beta_{\text{decr}}$; Go to 6;
4. If $\varepsilon^{(i)} > \alpha_{\text{incr}} \cdot \varepsilon_{\max}$ then $\boldsymbol{\delta}^{(i+1)} = \boldsymbol{\delta}^{(i)} \cdot \beta_{\text{incr}}$; Go to 6;
5. Set $\boldsymbol{\delta}^{(i+1)} = \boldsymbol{\delta}^{(i)}$;
6. END;

Here, α_{decr}, α_{incr}, β_{decr}, and β_{incr} are user-defined parameters (typical values: $\alpha_{\text{decr}} = 1$, $\alpha_{\text{incr}} = 2$, $\beta_{\text{decr}} = 5$, $\beta_{\text{incr}} = 2$). Our algorithm tightens the PE constraints if the approximation error is sufficiently small; otherwise, it loosens them.

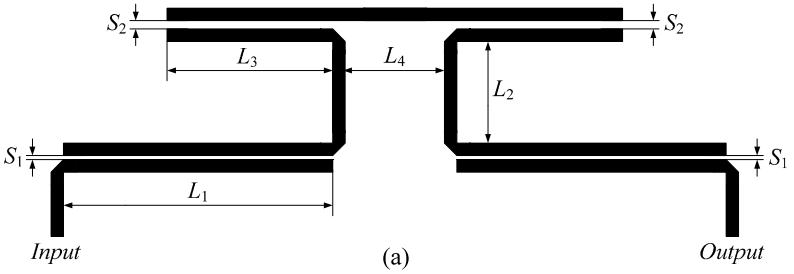
Note that constraint tightening improves the generalization capability of the surrogate model: the low approximation error $\varepsilon^{(i-1)} = \|\boldsymbol{R}_f(\boldsymbol{x}^{(i-1)}) - \boldsymbol{R}_c(\boldsymbol{x}^{(i-1)}, \boldsymbol{x}_p^{(i-1)})\|_p$ and $\varepsilon^{(i)} = \|\boldsymbol{R}_f(\boldsymbol{x}^{(i)}) - \boldsymbol{R}_c(\boldsymbol{x}^{(i)}, \boldsymbol{x}_p^{(i)})\|_p$ (satisfied when determining $\boldsymbol{x}_p^{(i-1)}$ and $\boldsymbol{x}_p^{(i)}$, respectively) makes it more likely to have $\|\boldsymbol{R}_f(\boldsymbol{x}^{(i-1)}) - \boldsymbol{R}_c(\boldsymbol{x}^{(i-1)}, \boldsymbol{x}_p^{(i)})\|_p$ small if $\boldsymbol{\delta}^{(i)}$ is reduced (because small $\|\boldsymbol{x}_p^{(i)} - \boldsymbol{x}_p^{(i-1)}\|_\infty \le \|\boldsymbol{\delta}^{(i)}\|_\infty$ implies the similarity of subsequent surrogate models).

The convergence properties of the SM algorithm can be explicitly controlled by constraining the surrogate optimization (2), which can be formulated as

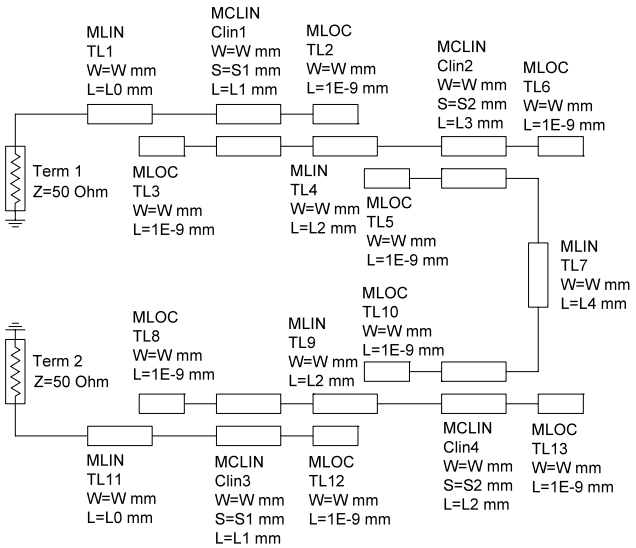
$$\boldsymbol{x}^{(i+1)} = \arg \min_{\boldsymbol{x}, \|\boldsymbol{x} - \boldsymbol{x}^{(i)}\| \le \delta^{(i)}} U\left(\boldsymbol{R}_c\left(\boldsymbol{x}, \boldsymbol{x}_p^{(i)}\right)\right), \tag{8}$$

where $\delta^{(i)} = \alpha \cdot \|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(i-1)}\|$ with $\alpha < 1$ (recommended values are $\alpha = 0.6$ to 0.9; values too small could result in premature convergence without finding a satisfactory design). Note that (8) is mostly used as a safeguard, because good convergence should be ensured by the constrained PE algorithm formulated above. On the other hand, constrained surrogate optimization is useful in finding a new design in the close neighborhood of the current design $\boldsymbol{x}^{(i)}$ (an over-flexible surrogate model may result in many designs that are good with respect to specification error, but are not necessarily close to $\boldsymbol{x}^{(i)}$).

It should be emphasized that the procedure described here is not related to the trust region approach [22] that has been used to safeguard convergence for SM algorithms [30]. The latter reduces the search range for the surrogate model. It rejects a new design if it does not bring sufficient improvement with respect to the fine model specification [32]. This increases the computational cost of SM optimization. According to the algorithm described here, the new design is never rejected: the generalization of the surrogate model is accommodated by the adaptive PE procedure. As indicated in [28], the constrained PE algorithm is expected to improve the convergence of the SM algorithm. The details can be found in [28].



(a)



(b)

Fig. 2 Coupled microstrip bandpass filter: (**a**) geometry [33], (**b**) coarse model (Agilent ADS)

3.3 Design Example: Coupled Microstrip Bandpass Filter

Consider the coupled microstrip bandpass filter [33] shown in Fig. 2(a). The design parameters are $\boldsymbol{x} = [L_1\ L_2\ L_3\ L_4\ S_1\ S_2]^T$ mm. The fine model is simulated in FEKO [34]. The coarse model, Fig. 2(b), is an equivalent circuit implemented in Agilent ADS [35]. The design specifications are $|S_{21}| \geq -3$ dB for 2.3 GHz $\leq \omega \leq$ 2.5 GHz, and $|S_{21}| \leq -20$ dB for 1.8 GHz $\leq \omega \leq$ 2.15 GHz and 2.65 GHz $\leq \omega \leq$ 3.0 GHz. The initial design is $\boldsymbol{x}^{(0)} = [24\ 4\ 14\ 12\ 0.2\ 0.1]^T$ mm.

The SM surrogate model has 14 preassigned parameters: $\varepsilon_{r1} = \varepsilon_{r.\mathrm{Clin1}} = \varepsilon_{r.\mathrm{Clin3}}$, $\varepsilon_{r2} = \varepsilon_{r.\mathrm{TL4}} = \varepsilon_{r.\mathrm{TL9}}$, $\varepsilon_{r3} = \varepsilon_{r.\mathrm{Clin2}} = \varepsilon_{r.\mathrm{Clin4}}$, $\varepsilon_{r4} = \varepsilon_{r.\mathrm{TL7}}$, $H_1 = H_{\mathrm{Clin1}} = H_{\mathrm{Clin3}}$, $H_2 = H_{\mathrm{TL4}} = H_{\mathrm{TL9}}$, $H_3 = H_{\mathrm{Clin2}} = H_{\mathrm{Clin4}}$, $H_4 = H_{\mathrm{TL7}}$, and dL_1, dL_2, dL_3, dL_4, dS_1, dS_2 (design variable perturbations). Thus, we have $\boldsymbol{x}_p = [\varepsilon_{r1}\ \varepsilon_{r2}\ \varepsilon_{r3}\ \varepsilon_{r4}\ H_1\ H_2\ H_3\ H_4\ dL_1\ dL_2\ dL_3\ dL_4\ dS_1\ dS_2]^T$. Symbols $\varepsilon_{r.\mathrm{Elem}}$ and

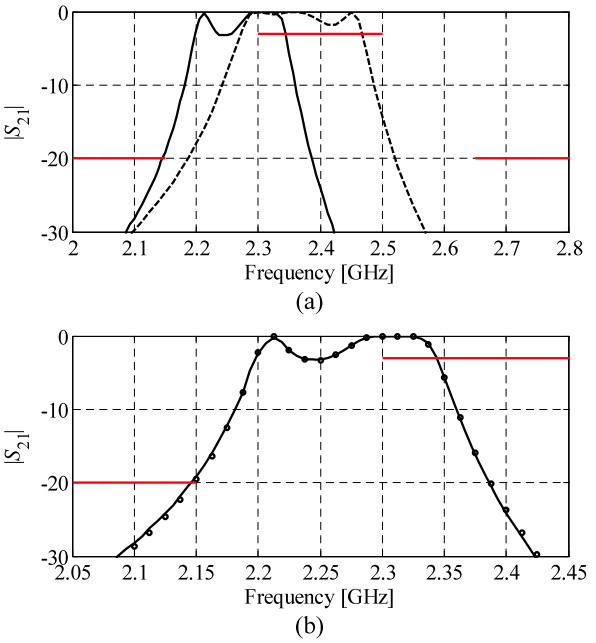


Fig. 3 Coupled microstrip bandpass filter: (**a**) responses of the fine model (*solid line*) and the coarse model (*dashed line*) at the initial design; (**b**) responses of the fine model (*solid line*) and the SM surrogate model (*circles*) at the initial design after PE. Note the excellent match between the models

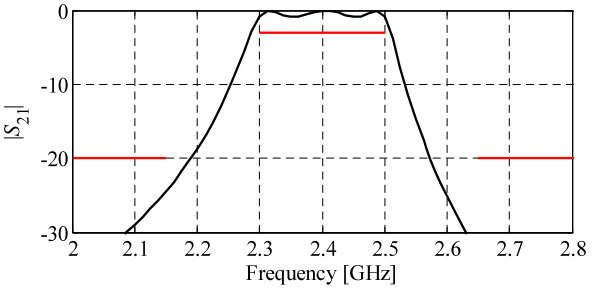


Fig. 4 Coupled microstrip bandpass filter: fine model response at the final design obtained

H_{Elem} refer to the dielectric constant (initial value 3.0) and substrate height (initial value 0.51 mm) of element *Elem*, respectively. Initial values for other parameters are zero. We take $\boldsymbol{\delta}^{(0)} = [1\ 1\ 1\ 1\ 0.3\ 0.3\ 0.3\ 0.3\ 0.4\ 0.4\ 0.4\ 0.4\ 0.02\ 0.02]^T$ (in respective units).

Figure 3(a) shows the fine and coarse model responses at the initial design. Thanks to the large number of surrogate model parameters, we obtain an excellent match to the fine model, as illustrated in Fig. 3(b). Figure 4 shows the fine model response at the final design.

Table 1 Coupled bandpass filter: optimization results

Algorithm	Specification error		Number of fine model evaluations
	Best found	Final	
Standard SM	−0.9 dB	−0.4 dB	21[a]
Trust region SM	−2.1 dB	−2.1 dB	21[b]
SM with constrained PE	−2.1 dB	−2.1 dB	12

[a]Algorithm terminated after 20 iterations without convergence

[b]Terminated after 21 fine model evaluations (good convergence pattern, but tolerance requirements not fulfilled yet)

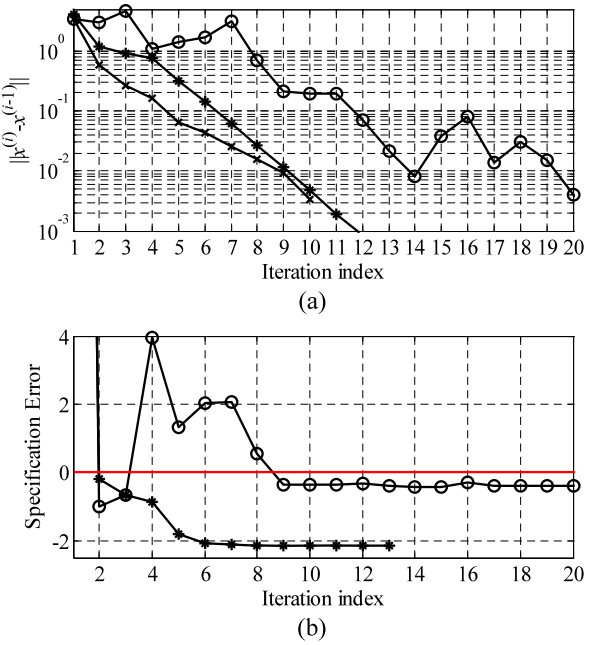


Fig. 5 Coupled microstrip bandpass filter: (**a**) convergence plot for the standard SM (o), trust region enhanced SM (×), and for the constrained algorithm (∗). Note that there is more than one fine model evaluation per iteration for the trust region algorithm (21 evaluations in total); (**b**) specification error versus iteration index for the standard SM (o) and for the constrained algorithm (∗)

Optimization results for standard and trust region enhanced SM as well as for the constrained algorithm of Sect. 3.2 are summarized in Table 1 as well as in Fig. 5. We observe that the standard algorithm exhibits features typical of SM: fast initial progress, then stagnation, as indicated by the lack of or slow convergence, and oscillation in the specification error values. On the other hand, the algorithm exhibits both a nice convergence pattern and consistent behavior with respect to specification error. The trust region convergence improves the convergence properties of the

algorithm, but at the expense of extra computational effort: any designs that do not reduce the specification error are rejected, so typically more than one fine model evaluation per iteration is required for this algorithm. The total optimization cost is almost twice as high as for the constrained algorithm.

4 Design of Antennas Using Space Mapping with Response Surface Approximation Coarse Models

In this section we deal with a specific modification of the SM algorithm that is suitable in situations when the underlying low-fidelity model is relatively expensive. This is particularly the case for antenna structures, where reliable equivalent circuit models are usually not available, and the only universal way to create a faster representation of the structure is coarse-discretization EM simulation.

4.1 Response Surface Approximation Coarse Models

As mentioned before, the coarse model is a critical component of successful SM optimization. The model should be physics-based; i.e., it should describe the same phenomena as the fine model. This ensures that the surrogate model constructed using $\boldsymbol{R}_c$ will have good prediction capability [15]. Also $\boldsymbol{R}_c$ should be computationally much cheaper than $\boldsymbol{R}_f$ so that the total costs of surrogate model optimization (2) and PE (4) problems are negligible. The preferred choice for the coarse model is therefore a circuit equivalent, e.g., one implemented in Agilent ADS [35]. Unfortunately, for many structures, particularly antennas, it is difficult to build a reliable circuit equivalent, or the circuit equivalent may be of insufficient accuracy, resulting in poor performance of the SM process.

In general, $\boldsymbol{R}_c$ can be implemented using the same EM solver as the one of the fine model by applying relaxed mesh requirements. However, the coarse-mesh model may have poor analytical properties (e.g., numerical noise, nondifferentiability and discontinuity of the response over the design variables) which make optimization of the surrogate difficult [11]. Also, it is not straightforward to find an appropriate trade-off between the model accuracy and evaluation time. The rule of thumb is that the evaluation time of $\boldsymbol{R}_c$ should be at least two orders of magnitude smaller than that of $\boldsymbol{R}_f$ in order to make the overhead of solving (2) and (4) reasonably small.

We mention that coarse mesh is not the only different feature of the coarse model compared to the fine model. Other options include: (i) a reduced number of cells in the perfectly matched layer absorbing boundary conditions as well as a reduced distance from the simulated structure to the absorbing boundary conditions, in the case of finite-volume EM simulators [36–38]; (ii) the lower order of the basis functions, in the case of finite-element and integral equation solvers [37, 38]; (iii) simplified

excitation [38, 39], e.g., the discrete source of the coarse model versus the waveguide port of the fine model; (iv) zero thickness of metallization; (v) the use of a perfect electric conductor in place of finite-conductivity metals.

In this chapter, a coarse model is built using a functional approximation of data obtained with the same EM simulator as the fine model but with a much coarser mesh. This coarse-discretization EM-based model will be denoted as $\boldsymbol{R}_{\rm cd}$. Note that the functional approximation model would normally be set up using sampled fine model data. Here, in order to reduce the computational overhead, the surrogate is constructed using its simplified representation, $\boldsymbol{R}_{\rm cd}$.

A variety of function approximation methods are available, including polynomial approximation [5], neural networks [40–44], kriging [5, 45, 46], multidimensional Cauchy approximation [47], or support vector regression [48]. Here, the coarse model is constructed using kriging interpolation. We made this choice not only because kriging is a reliable and popular technique [5], but also because the available Matlab kriging toolbox, DACE [49], allows us to configure kriging-based models in an efficient way. For brevity, we omit a formulation of the details of kriging. The interested reader can find them in the literature (e.g., [5] or [9]).

Response surface approximation in general (and kriging interpolation in particular) as a method of generating the coarse model for the SM algorithm has a number of advantages: (i) the resulting model is computationally cheap, smooth, and therefore, easy to optimize; (ii) there is no need for an equivalent circuit model, and, consequently, no extra simulation software is needed; the SM algorithm implementation is simpler and exploits a single EM solver; (iii) it is possible to apply SM for antenna design problems where finding reliable and fast coarse models is difficult or impossible; (iv) the initial design obtained through optimization of the coarse-mesh EM model is usually better than the initial design that could be obtained by other methods.

4.2 Design Optimization Process

The design optimization procedure can be summarized as follows (Fig. 6):

1. Take initial design $\boldsymbol{x}^{\rm init}$;
2. Find the starting point $\boldsymbol{x}^{(0)}$ for SM algorithm by optimizing the coarse-discretization model $\boldsymbol{R}_{\rm cd}$;
3. Allocate N base designs, $X_B = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N\}$;
4. Evaluate $\boldsymbol{R}_{\rm cd}$ at each design $\boldsymbol{x}^j$, $j = 1, 2, \ldots, N$;
5. Build the coarse model $\boldsymbol{R}_c$ as a kriging interpolation of data pairs $\{(\boldsymbol{x}^j, \boldsymbol{R}_{\rm cd}(\boldsymbol{x}^j))\}_{j=1,\ldots,N}$;
6. Set $i = 0$;
7. Evaluate the fine model $\boldsymbol{R}_f$ at $\boldsymbol{x}^{(i)}$;
8. Construct the space mapping surrogate model $\boldsymbol{R}_s^{(i)}$ as in (3) and (4);
9. Find a new design $\boldsymbol{x}^{(i+1)}$ as in (2);

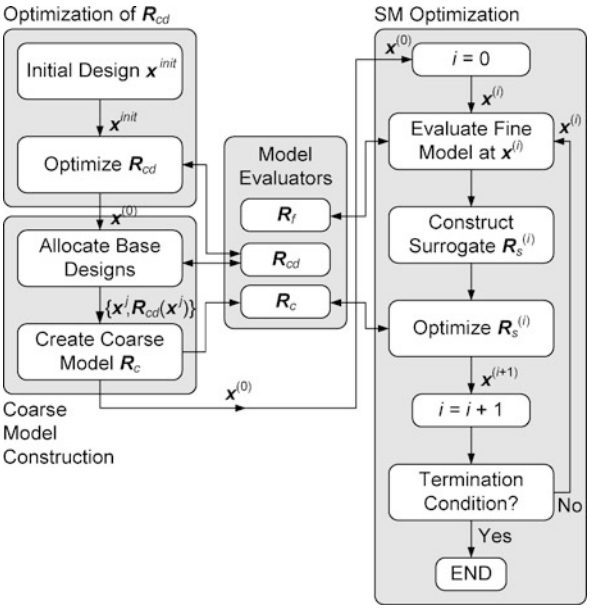


Fig. 6 Flowchart of the design optimization procedure exploiting a response-surface-approximation-based coarse model and SM as the main optimization engine [9]

10. Set $i = i + 1$;
11. If the termination condition is not satisfied go to 7;
12. END

The first phase of the design process is to find an optimized design of the coarse-discretization model. The optimum of $\boldsymbol{R}_{\text{cd}}$ is usually the best design we can get at a reasonably low computational cost. This cost can be further reduced by relaxing tolerance requirements while searching for $\boldsymbol{x}^{(0)}$: due to a limited accuracy of $\boldsymbol{R}_{\text{cd}}$ it is sufficient to find only a rough approximation of its optimum. Steps 3–5 describe the construction of the kriging-based coarse model. Steps 6–12 describe the flow of the SM algorithm. In principle, any SM surrogate model can be used except one from implicit SM, as the kriging-based coarse model normally does not inherit pre-assigned parameters from the coarse-discretization EM model. Here, the algorithm is terminated when no improvement of the fine model objective function is obtained in a given iteration. In general, the algorithm can be embedded in the trust region framework [22] for improved convergence.

The mesh density for the coarse-discretization model $\boldsymbol{R}_{\text{cd}}$ should be adjusted so that its evaluation time is substantially smaller than that of the fine model and, at the same time, its accuracy is still decent. Typically, if $\boldsymbol{R}_{\text{cd}}$ is set up so that it is 20 to 60 times faster than $\boldsymbol{R}_f$, its accuracy is acceptable for the purpose of the design procedure. The coarse model is created in the neighborhood X_N of $\boldsymbol{x}^{(0)}$, the approximate optimum of the coarse-discretization model $\boldsymbol{R}_{\text{cd}}$. The relative size of this neighborhood depends on the sensitivity of the antenna response to design variables as well

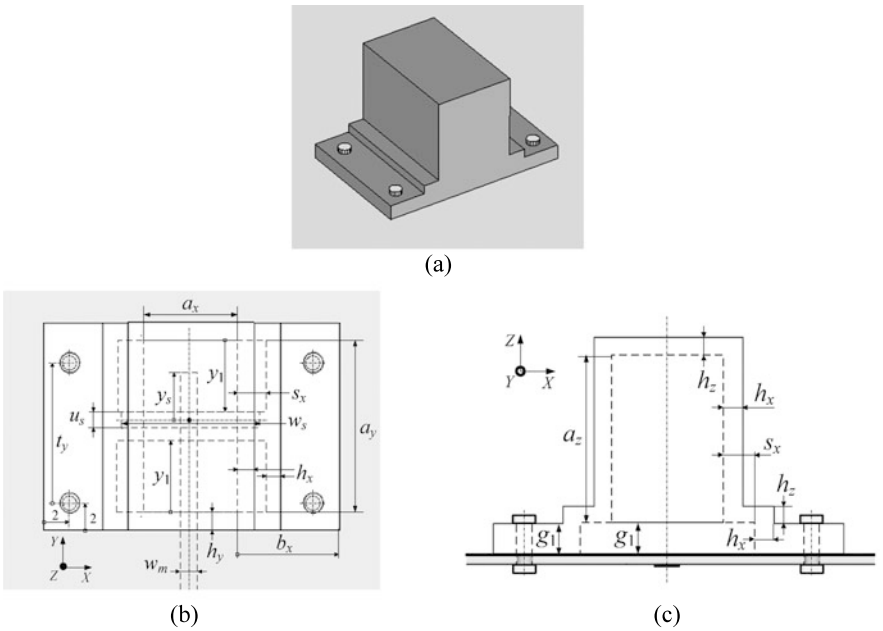


Fig. 7 DRA: (**a**) 3D view, (**b**) top view, and (**c**) front view; substrate shown transparent

as the discrepancy between the fine and coarse-discretization models, and may vary from a few to 20 percent. The number of base designs N depends on the problem dimensionality; typical values are 50 to 200. The base points should be allocated as uniformly as possible in X_N. Here, we use a modified Latin hypercube sampling [50] algorithm that gives a fairly uniform distribution of samples regardless of the number of design variables.

4.3 Design Example: Dielectric Resonator Antenna

Consider a rectangular DRA [51]; see Fig. 7 for its geometry. The DRA comprises a rectangular dielectric resonator (DR) estimated to operate at the perturbed $TE_{\delta 11}$ mode [51], supporting RO4003C [52] slabs, and polycarbonate housing. The housing is fixed to the circuit board with four through M1 bolts. The DRA is energized with a 50 Ω microstrip through a slot made in the metal ground. The substrate is 0.5 mm thick RO4003C. The design specifications are $|S_{11}| \leq -15$ dB for 5.1–5.9 GHz; also the DRA is required to have an antenna gain of better than 5 dBi for the zero zenith angle over the bandwidth of interest.

There are nine design variables: $\boldsymbol{x} = [a_x\ a_y\ a_z\ a_{y0}\ u_s\ w_s\ y_s\ g_1\ y_1]^T$, where a_x, a_y, and a_z are the dimensions of the DR, a_{y0} stands for the offset of the DR center relative to the slot center (marked by a black dot in Fig. 7(b)) in the Y-direction, u_s

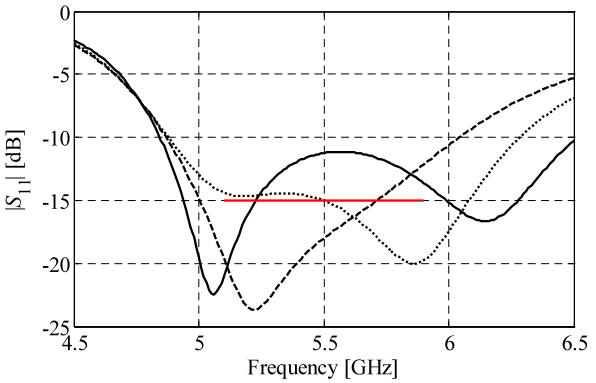


Fig. 8 DRA, $|S_{11}|$ versus frequency: fine model $\boldsymbol{R}_f$ at the initial design (- - -), optimized coarse-discretization model $\boldsymbol{R}_{\text{cd}}$ ($\cdot\ \cdot\ \cdot$), and $\boldsymbol{R}_f$ at the optimum of $\boldsymbol{R}_{\text{cd}}$ ($-$)

and w_s are the slot dimensions, y_s is the length of the microstrip stub, and g_1 and y_1 are the slab dimensions. The relative permittivity and loss tangent of the DR are 10 and 1e-4, respectively, at 6.5 GHz.

The width of the microstrip signal trace is 1.15 mm. Metallization of the trace and ground is done with 50 μm copper. The relative permittivity and loss tangent of the polycarbonate housing are 2.8 and 0.01 at 6.5 GHz, respectively. DRA models are defined with the CST MWS [39], and the built-in single-pole Debye model is used for all dielectrics to describe their dispersion properties. Other dimensions are fixed as follows: $h_x = h_y = h_z = 1$, $b_x = 7.5$, $s_x = 2$, and $t_y = a_y - a_{y0} - 1$, all in millimeters.

The initial design is $\boldsymbol{x}^{\text{in}} = [8.000\ 14.000\ 9.000\ 0\ 1.750\ 10.000\ 3.000\ 1.500\ 6.000]^T$ mm. The fine (1,099,490 mesh cells at $\boldsymbol{x}^{\text{in}}$) and coarse-discretization (26,796 mesh cells at $\boldsymbol{x}^{\text{in}}$) antenna models are evaluated with CST MWS transient solver in 2,175 and 42 s, respectively.

A coarse-discretization model optimum is $\boldsymbol{x}^{(0)} = [7.444\ 13.556\ 9.167\ 0.250\ 1.750\ 10.500\ 2.500\ 1.500\ 6.000]^T$ mm. Figure 8 shows the fine model reflection response at the initial design as well as that of the fine and coarse-discretization model $\boldsymbol{R}_{\text{cd}}$ at $\boldsymbol{x}^{(0)}$. The kriging coarse model is set up using 200 samples of $\boldsymbol{R}_{\text{cd}}$ allocated in the vicinity of $\boldsymbol{x}^{(0)}$ of size $[0.5\ 0.5\ 0.5\ 0.25\ 0.5\ 0.25\ 0.25\ 0.25\ 0.5]^T$ mm.

The final design, $\boldsymbol{x}^{(4)} = [7.556\ 13.278\ 9.630\ 0.472\ 1.287\ 10.593\ 2.667\ 1.722\ 6.482]^T$ mm, is obtained after four SM iterations; its reflection response is shown in Fig. 9. The far-field response of the final design is shown in Fig. 10. For the bandwidth of interest, the peak gain is above 5 dBi, and the back radiation level is below -14 dB (relative to the maximum). All responses shown include the effect of the 25 mm input microstrip. The surrogate model used by the optimization algorithm exploited input and output SM of the form $\boldsymbol{R}_s(\boldsymbol{x}) = \boldsymbol{R}_c(\boldsymbol{x} + \boldsymbol{c}) + \boldsymbol{d}$. The optimization costs are summarized in Table 2. The total design time corresponds to about 11 evaluations of the fine model.

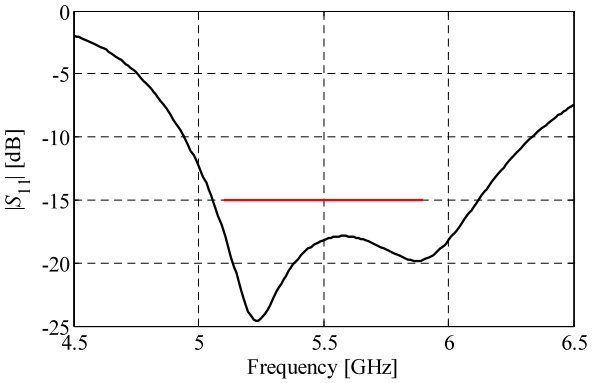


Fig. 9 DRA, $|S_{11}|$ versus frequency: $\boldsymbol{R}_f$ at the final design

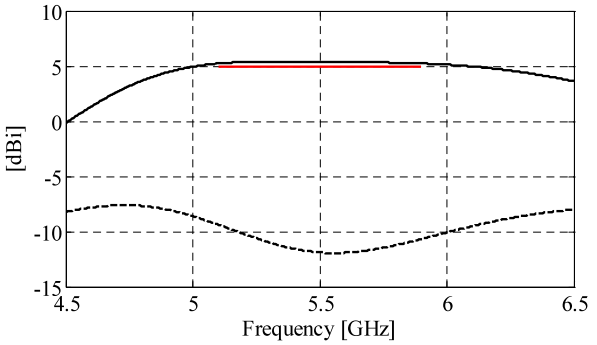


Fig. 10 DRA, realized gain versus frequency: (—) is for the zero zenith angle ($\theta = 0^0$); (- - -) is back radiation for $\theta = 180°$. Here, only θ-polarization ($\phi = 90°$) contributes to the gain for the listed directions

Table 2 DRA: optimization cost

Algorithm component	Number of model evaluations	CPU time	
		Absolute	Relative to $\boldsymbol{R}_f$
Optimization of $\boldsymbol{R}_{\text{cd}}$	$150 \times \boldsymbol{R}_{\text{cd}}$	105 min	2.9
Setting up $\boldsymbol{R}_c$	$200 \times \boldsymbol{R}_{\text{cd}}$	140 min	3.9
Evaluation of $\boldsymbol{R}_f$	$4 \times \boldsymbol{R}_f$	145 min	4.0
Total cost	N/A	390 min	10.8

5 Space Mapping with Adjoint Sensitivity

In this section, we discuss the use of adjoint sensitivity [53] to enhance conventional SM optimization algorithms. More specifically, adjoint sensitivity is exploited to:

(i) speed up the surrogate model optimization process (2), (ii) speed up the PE process (4), and (iii) improve the matching between the surrogate and the high-fidelity model. Due to (i) and (ii), both PE and surrogate model optimization can be performed using a small number of low-fidelity model evaluations, which allows us to utilize coarse-discretization EM coarse models. This widens SM applications as well as improving SM performance, as low-fidelity SM models are normally quite accurate. Adjoint sensitivities are also used to match both the responses and first-order derivatives of the surrogate and the high-fidelity model, which improves the performance and convergence properties of the SM algorithm [54]. An illustrative example is provided.

5.1 Low-Fidelity and Surrogate Models

SM with adjoint sensitivity follows the generic SM scheme (2). As usual, the surrogate model is constructed using the underlying low-fidelity (or coarse) model $\boldsymbol{R}_c$, which is a simplified representation of the high-fidelity one. Here, we focus on coarse-discretization EM models, as they allow us to extend the applicability of SM to all conceivable microwave structures. Coarse-discretization EM models are typically accurate but relatively expensive; therefore, the SM surrogate exploited here is based on input and output SM [21] of the form:

$$\boldsymbol{R}_s^{(i)}(\boldsymbol{x}) = \boldsymbol{R}_c\big(\boldsymbol{x} + \boldsymbol{c}^{(i)}\big) + \boldsymbol{d}^{(i)} + \boldsymbol{E}^{(i)}\big(\boldsymbol{x} - \boldsymbol{x}^{(i)}\big). \tag{9}$$

Here, only the input SM vector $\boldsymbol{c}^{(i)}$ is obtained through the nonlinear PE process

$$\boldsymbol{c}^{(i)} = \arg\min_{\boldsymbol{c}} \big\| \boldsymbol{R}_f\big(\boldsymbol{x}^{(i)}\big) - \boldsymbol{R}_c\big(\boldsymbol{x}^{(i)} + \boldsymbol{c}\big) \big\|. \tag{10}$$

Output SM parameters are calculated as

$$\boldsymbol{d}^{(i)} = \boldsymbol{R}_f\big(\boldsymbol{x}^{(i)}\big) - \boldsymbol{R}_c\big(\boldsymbol{x}^{(i)} + \boldsymbol{c}^{(i)}\big) \tag{11}$$

and

$$\boldsymbol{E}^{(i)} = \boldsymbol{J}_{\boldsymbol{R}_f}\big(\boldsymbol{x}^{(i)}\big) - \boldsymbol{J}_{\boldsymbol{R}_c}\big(\boldsymbol{x}^{(i)} + \boldsymbol{c}^{(i)}\big), \tag{12}$$

where $\boldsymbol{J}$ denotes the Jacobian of the respective model obtained using adjoint sensitivities. Formulation (2)–(5) ensures zero- and first-order consistency [55] between the surrogate and the fine model, i.e., $\boldsymbol{R}_s^{(i)}(\boldsymbol{x}^{(i)}) = \boldsymbol{R}_f(\boldsymbol{x}^{(i)})$ and $\boldsymbol{J}_{\boldsymbol{R}_s^{(i)}}(\boldsymbol{x}^{(i)}) = \boldsymbol{J}_{\boldsymbol{R}_f}(\boldsymbol{x}^{(i)})$, which substantially improves the ability of the SM algorithm to quickly locate the high-fidelity model optimum [55]. Here, the algorithm (2) is embedded in the trust region framework [30]; i.e., we have $\boldsymbol{x}^{(i+1)} = \arg\min\{\|\boldsymbol{x} - \boldsymbol{x}^{(i)}\| \le \delta^{(i)} : U(\boldsymbol{R}_s^{(i)}(\boldsymbol{x}))\}$, where the trust region radius $\delta^{(i)}$ is updated using classical rules [30]. Assuming first-order consistency and smoothness of $\boldsymbol{R}_s^{(i)}(\boldsymbol{x})$, this ensures convergence to the local $\boldsymbol{R}_f$ optimum.

5.2 Fast Parameter Extraction and Surrogate Optimization Using Adjoint Sensitivities

To speed up the PE process (4), we exploit adjoint sensitivities. We use a simple trust-region-based [22] algorithm, where the approximate solution $\boldsymbol{c}^{(i.k+1)}$ of $\boldsymbol{c}^{(i)}$ is found as (k is the iteration index for PE process (13))

$$\boldsymbol{c}^{(i.k+1)} = \arg\min_{\|\boldsymbol{c}-\boldsymbol{c}^{(i.k)}\| \le \delta_{\mathrm{PE}}^{(k)}} \left\| \boldsymbol{R}_f\left(\boldsymbol{x}^{(i)}\right) - \boldsymbol{L}_{c.c}^{(i.k)}(\boldsymbol{c}) \right\|, \tag{13}$$

where $\boldsymbol{L}_{c.c}^{(i.k)}(\boldsymbol{c}) = \boldsymbol{R}_c(\boldsymbol{x}^{(i)} + \boldsymbol{c}^{(i.k)}) + \boldsymbol{J}_{\boldsymbol{R}_c}(\boldsymbol{x}^{(i)} + \boldsymbol{c}^{(i.k)}) \cdot (\boldsymbol{c} - \boldsymbol{c}^{(i.k)})$ is a linear approximation of $\boldsymbol{R}_c(\boldsymbol{x}^{(i)} + \boldsymbol{c})$ at $\boldsymbol{c}^{(i.k)}$. The trust region radius $\delta_{\mathrm{PE}}^{(k)}$ is updated according to standard rules [22]. PE is terminated upon convergence or exceeding the maximum number of coarse model evaluations (here, the limit is set to 5, which is sufficient when using adjoint sensitivity).

Adjoint sensitivities are also utilized to lower the cost of surrogate model optimization. Similarly to (13), we use a trust-region-based algorithm that produces a sequence of approximations $\boldsymbol{x}^{(i+1.k)}$ of the solution $\boldsymbol{x}^{(i+1)}$ to (2) as follows (k is the iteration index for surrogate model optimization process (14)):

$$\boldsymbol{x}^{(i+1.k+1)} = \arg\min_{\|\boldsymbol{x}-\boldsymbol{x}^{(i+1.k)}\| \le \delta_{\mathrm{SO}}^{(k)}} U\left(\boldsymbol{L}_{c.x}^{(i.k)}(\boldsymbol{x})\right), \tag{14}$$

where $\boldsymbol{L}_{c.x}^{(i.k)}(\boldsymbol{x}) = \boldsymbol{R}_s^{(i)}(\boldsymbol{x}^{(i+1.k)} + \boldsymbol{c}^{(i)}) + \boldsymbol{J}_{\boldsymbol{R}_s^{(i)}}(\boldsymbol{x}^{(i+1.k)} + \boldsymbol{c}^{(i)}) \cdot (\boldsymbol{x} - \boldsymbol{x}^{(i+1.k)})$ is a linear approximation of $\boldsymbol{R}_s^{(i)}(\boldsymbol{x} + \boldsymbol{c}^{(i)})$ at $\boldsymbol{x}^{(i+1.k)}$. The trust region radius $\delta_{\mathrm{SO}}^{(k)}$ is updated according to standard rules [54]. Typically, due to adjoint sensitivities, surrogate model optimization requires only a few evaluations of the coarse model $\boldsymbol{R}_c$. Note that the sensitivities of the surrogate model can be calculated using the sensitivities of both $\boldsymbol{R}_f$ and $\boldsymbol{R}_c$ as follows: $\boldsymbol{J}_{\boldsymbol{R}_s^{(i)}}(\boldsymbol{x} + \boldsymbol{c}^{(i)}) = \boldsymbol{J}_{\boldsymbol{R}_c}(\boldsymbol{x} + \boldsymbol{c}^{(i)}) + [\boldsymbol{J}_{\boldsymbol{R}_f}(\boldsymbol{x}^{(i)}) - \boldsymbol{J}_{\boldsymbol{R}_c}(\boldsymbol{x}^{(i)} + \boldsymbol{c}^{(i)})]$.

5.3 Design Example: Dielectric Resonator Filter

Consider the dielectric resonator filter [56] shown in Fig. 11. The design variables are $\boldsymbol{x} = [u_1\ u_2\ u_3\ v_1\ v_2]^T$ mm, and the relative permittivity of the dielectric resonators (DR) is $\varepsilon_{r1} = 38$. The relative permittivity of the DR supports and the coax filling is $\varepsilon_{r2} = 2.1$. The materials are considered lossless. Here $h_3 = h_4 = h_5 = 1$ mm, $r_3 = r_2$ (the inner radius of the DR), and $r_4 = r_2 + 1$ mm, and $\rho_1 = 2.05$ mm. Other dimensions are fixed as in [56].

The filter models are simulated by the CST MWS transient solver [39]. The coarse-discretization model $\boldsymbol{R}_{\mathrm{cd}}$ comprises 38,220 hexahedral cells and runs for 1 min, and the fine model $\boldsymbol{R}_f$ comprises 333,558 cells and runs for 28 min, both at the initial design which is $\boldsymbol{x}^{(\mathrm{init})} = [8\ 12\ 25\ 25\ -10]^T$ mm. The design specifications are $|S_{21}| \ge -0.75$ dB for 4.52 GHz $\le \omega \le$ 4.54 GHz, and $|S_{21}| \le -20$ dB for 4.4 GHz $\le \omega \le$ 4.47 GHz and 4.59 GHz $\le \omega \le$ 4.63 GHz.

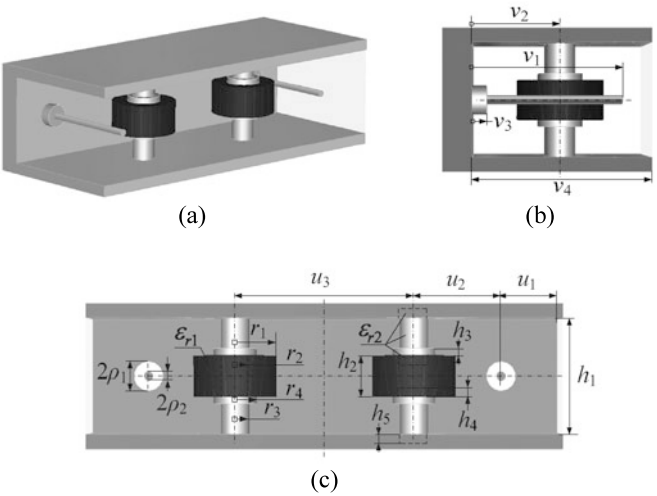


Fig. 11 Dielectric resonator filter [56]: (**a**) 3D view, (**b**) side view, and (**b**) front view. Front and side metal walls of the waveguide section are shown transparent

Table 3 Dielectric resonator filter: optimization results

Algorithm component	Number of model evaluations[a]	CPU time	
		Absolute [min]	Relative to $\boldsymbol{R}_f$
Evaluation of $\boldsymbol{R}_c$	80	80	2.9
Evaluation of $\boldsymbol{R}_f$	11	308	11
Total cost[a]	N/A	398	**13.9**

[a]Excludes $\boldsymbol{R}_f$ evaluation at the initial design

The filter was optimized using the algorithm of Sects. 5.1 and 5.2. Figure 12(a) shows the responses of $\boldsymbol{R}_f$ and $\boldsymbol{R}_c$ at $\boldsymbol{x}^{\text{init}}$. Figure 12(b) shows the response of the fine model at the final design $\boldsymbol{x}^{(*)} = [7.28\ 11.06\ 24.81\ 26.00\ -11.07]^T$ mm obtained after ten SM iterations. Table 3 summarizes the total optimization cost. Note that using adjoint sensitivities allows us to greatly reduce the number of both fine and coarse model evaluations in the design process. The average cost of the PE and surrogate optimization processes is only about 5 evaluations of $\boldsymbol{R}_c$. The evolution of the specification error is shown in Fig. 13.

6 Conclusion

In this chapter, a number of design examples concerning various microwave components have been presented, including microstrip filters and planar antennas, as well as transition structures. In all cases, the surrogate-based techniques presented

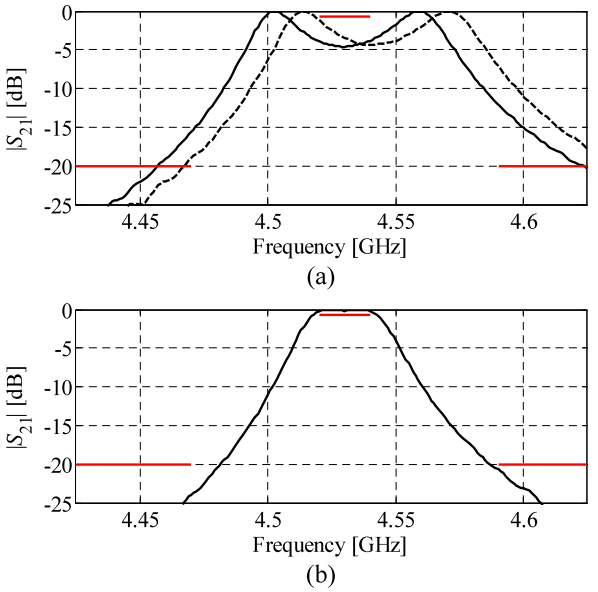


Fig. 12 Dielectric resonator filter: (**a**) responses of $\boldsymbol{R}_f$ (—) and $\boldsymbol{R}_c$ (- - -) at the initial design $\boldsymbol{x}^{\text{init}}$, (**b**) response of $\boldsymbol{R}_f$ (—) at the final design

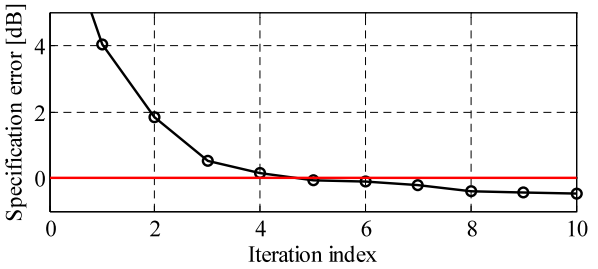


Fig. 13 Dielectric resonator filter: minimax specification error versus SM iteration index

here have been employed as optimization engines. The results presented here indicate that the surrogate-based optimization methods make the simulation-driven microwave design feasible and efficient, in terms of both the quality of the final design and the computational cost. In most cases, the design cost corresponds to a few high-fidelity electromagnetic simulations of the microwave structure under consideration, typically comparable to the number of design variables. While this kind of performance is definitely appealing, improved robustness and reliability as well as availability through commercial software packages are needed to make the surrogate-based techniques widely accepted by the microwave engineering community. Therefore, a substantial research effort in this area is expected in the years to come.

References

1. Bandler, J.W., Biernacki, R.M., Chen, S.H., Grobelny, P.A., Hemmers, R.H.: Space mapping technique for electromagnetic optimization. IEEE Trans. Microw. Theory Tech. **4**, 536–544 (1994)
2. Bakr, M.H., Bandler, J.W., Georgieva, N.K., Madsen, K.: A hybrid aggressive space-mapping algorithm for EM optimization. IEEE Trans. Microw. Theory Tech. **47**, 2440–2449 (1999)
3. Bakr, M.H., Bandler, J.W., Biernacki, R.M., Chen, S.H., Madsen, K.: A trust region aggressive space mapping algorithm for EM optimization. IEEE Trans. Microw. Theory Tech. **46**, 2412–2425 (1998)
4. Amari, S., LeDrew, C., Menzel, W.: Space-mapping optimization of planar coupled-resonator microwave filters. IEEE Trans. Microw. Theory Tech. **54**, 2153–2159 (2006)
5. Queipo, N.V., Haftka, R.T., Shyy, W., Goel, T., Vaidynathan, R., Tucker, P.K.: Surrogate-based analysis and optimization. Prog. Aerosp. Sci. **41**, 1–28 (2005)
6. Forrester, A.I.J., Keane, A.J.: Recent advances in surrogate-based optimization. Prog. Aerosp. Sci. **45**, 50–79 (2009)
7. Koziel, S., Echeverría-Ciaurri, D., Leifsson, L.: Surrogate-based methods. In: Koziel, S., Yang, X.S. (eds.) Computational Optimization, Methods and Algorithms. Studies in Computational Intelligence, pp. 33–60. Springer, Berlin (2011)
8. Bandler, J.W., Cheng, Q.S., Dakroury, S.A., Mohamed, A.S., Bakr, M.H., Madsen, K., Sondergaard, J.: Space mapping: the state of the art. IEEE Trans. Microw. Theory Tech. **52**, 337–361 (2004)
9. Koziel, S., Ogurtsov, S.: Rapid design optimization of antennas using space mapping and response surface approximation models. Int. J. RF Microw. Comput.-Aided Eng. **21**, 611–621 (2011)
10. Ouyang, J., Yang, F., Zhou, H., Nie, Z., Zhao, Z.: Conformal antenna optimization with space mapping. J. Electromagn. Waves Appl. **24**, 251–260 (2010)
11. Zhu, J., Bandler, J.W., Nikolova, N.K., Koziel, S.: Antenna optimization through space mapping. IEEE Trans. Antennas Propag. **55**, 651–658 (2007)
12. Schantz, H.: The Art and Science of Ultrawideband Antennas. Artech House, Boston (2005)
13. Wu, K.: Substrate integrated circuits (SiCs)—a new paradigm for future GHz and THz electronic and photonic systems. IEEE Circuits Syst. Soc. Newsl. **3**, 1 (2009)
14. Koziel, S., Bandler, S.W., Madsen, K.: A space mapping framework for engineering optimization: theory and implementation. IEEE Trans. Microw. Theory Tech. **54**, 3721–3730 (2006)
15. Koziel, S., Bandler, J.W., Madsen, K.: Quality assessment of coarse models and surrogates for space mapping optimization. Optim. Eng. **9**, 375–391 (2008)
16. Bandler, J.W., Cheng, Q.S., Nikolova, N.K., Ismail, M.A.: Implicit space mapping optimization exploiting preassigned parameters. IEEE Trans. Microw. Theory Tech. **52**, 378–385 (2004)
17. Koziel, S., Bandler, J.W.: Space-mapping optimization with adaptive surrogate model. IEEE Trans. Microw. Theory Tech. **55**, 541–547 (2007)
18. Nocedal, J., Wright, S.: Numerical Optimization, 2nd edn. Springer, Berlin (2006)
19. Kolda, T.G., Lewis, R.M., Torczon, V.: Optimization by direct search: new perspectives on some classical and modern methods. SIAM Rev. **45**, 385–482 (2003)
20. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Upper Saddle River (1989)
21. Koziel, S., Cheng, Q.S., Bandler, J.W.: Space mapping. IEEE Microw. Mag. **9**, 105–122 (2008)
22. Conn, A.R., Scheinberg, K., Vicente, L.N.: Introduction to Derivative-Free Optimization. MPS-SIAM Series on Optimization. MPS-SIAM, Philadelphia (2009)
23. Alexandrov, N.M., Dennis, J.E., Lewis, R.M., Torczon, V.: A trust region framework for managing use of approximation models in optimization. Struct. Multidiscip. Optim. **15**, 16–23 (1998)

24. Booker, A.J., Dennis, J.E. Jr., Frank, P.D., Serafini, D.B., Torczon, V., Trosset, M.W.: A rigorous framework for optimization of expensive functions by surrogates. Struct. Optim. **17**, 1–13 (1999)
25. Amari, S., LeDrew, C., Menzel, W.: Space-mapping optimization of planar coupled-resonator microwave filters. IEEE Trans. Microw. Theory Tech. **54**, 2153–2159 (2006)
26. Crevecoeur, G., Sergeant, P., Dupre, L., Van de Walle, R.: Two-level response and parameter mapping optimization for magnetic shielding. IEEE Trans. Magn. **44**, 301–308 (2008)
27. Koziel, S., Cheng, Q.S., Bandler, J.W.: Implicit space mapping with adaptive selection of preassigned parameters. IET Microw. Antennas Propag. **4**, 361–373 (2010)
28. Koziel, S., Bandler, J.W., Cheng, Q.S.: Constrained parameter extraction for microwave design optimization using implicit space mapping. IET Microw. Antennas Propag. **5**, 1156–1163 (2011)
29. Koziel, S., Bandler, J.W.: Coarse and surrogate model assessment for engineering design optimization with space mapping. In: IEEE MTT-S Int. Microwave Symp. Dig., pp. 107–110 (2007)
30. Koziel, S., Bandler, J.W., Cheng, Q.S.: Robust trust-region space-mapping algorithms for microwave design optimization. IEEE Trans. Microw. Theory Tech. **58**, 2166–2174 (2010)
31. Koziel, S., Bandler, J.W., Cheng, Q.S.: Adaptively constrained parameter extraction for robust space mapping optimization of microwave circuits. In: IEEE MTT-S Int. Microwave Symp. Dig., pp. 205–208 (2010)
32. Koziel, S., Bandler, J.W., Madsen, K.: Space-mapping based interpolation for engineering optimization. IEEE Trans. Microw. Theory Tech. **54**, 2410–2421 (2006)
33. Lee, H.-M., Tsai, C.-M.: Improved coupled-microstrip filter design using effective even-mode and odd-mode characteristic impedances. IEEE Trans. Microw. Theory Tech. **53**, 2812–2818 (2005)
34. FEKO User's Manual. Suite 6.0. EM Software & Systems-S.A. (Pty) Ltd, 32 Techno Lane, Technopark, Stellenbosch, 7600, South Africa (2010)
35. Agilent ADS, Version 2011, Agilent Technologies, 395 Page Mill Road, Palo Alto, CA, 94304 (2011)
36. Kempbel, L.C.: Computational electromagnetics for antennas. In: Volakis, J.L. (ed.) Antenna Engineering Handbook, 4th edn. McGraw-Hill, New York (2007)
37. Taflove, A., Hagness, S.C.: Computational Electrodynamics: The Finite-Difference Time-Domain Method, 3rd edn. Artech House, Boston (2005)
38. Jin, J.-M.: The Finite Element Method in Electromagnetics, 2nd edn. Wiley-IEEE Press, New York (2002)
39. CST Microwave Studio, ver. 2012. CST AG, Bad Nauheimer Str. 19, D-64289 Darmstadt, Germany (2012)
40. Kabir, H., Wang, Y., Yu, M., Zhang, Q.J.: Neural network inverse modeling and applications to microwave filter design. IEEE Trans. Microw. Theory Tech. **56**, 867–879 (2008)
41. Murphy, E.K., Yakovlev, V.V.: Neural network optimization of complex microwave structures with a reduced number of full-wave analyses. Int. J. RF Microw. Comput.-Aided Des. **21** (2010)
42. Gutiérrez-Ayala, V., Rayas-Sánchez, J.E.: Neural input space mapping optimization based on nonlinear two-layer perceptrons with optimized nonlinearity. Int. J. RF Microw. Comput.-Aided Eng. **20**, 512–526 (2010)
43. Tighilt, Y., Bouttout, F., Khellaf, A.: Modeling and design of printed antennas using neural networks. Int. J. RF Microw. Comput.-Aided Eng. **21**, 228–233 (2011)
44. Kabir, H., Yu, M., Zhang, Q.J.: Recent advances of neural network-based EM-CAD. Int. J. RF Microw. Comput.-Aided Eng. **20**, 502–511 (2010)
45. Siah, E.S., Ozdemir, T., Volakis, J.L., Papalambros, P., Wiese, R.: Fast parameter optimization using kriging metamodeling [antenna EM modeling/simulation]. In: IEEE Antennas and Prop. Int. Symp, pp. 76–79 (2003)
46. Siah, E.S., Sasena, M., Volakis, J.L., Papalambros, P.Y., Wiese, R.W.: Fast parameter optimization of large-scale electromagnetic objects using DIRECT with kriging metamodeling.

IEEE Trans. Microw. Theory Tech. **52**, 276–285 (2004)
47. Shaker, G.S.A., Bakr, M.H., Sangary, N., Safavi-Naeini, S.: Accelerated antenna design methodology exploiting parameterized Cauchy models. Prog. Electromagn. Res. **18**, 279–309 (2009)
48. Xia, L., Meng, J., Xu, R., Yan, B., Guo, Y.: Modeling of 3-D vertical interconnect using support vector machine regression. IEEE Microw. Wirel. Compon. Lett. **16**, 639–641 (2006)
49. Lophaven, S.N., Nielsen, H.B., Søndergaard, J.: DACE: a Matlab kriging toolbox. Technical University of Denmark (2002)
50. Beachkofski, B., Grandhi, R.: Improved distributed hypercube sampling. American Institute of Aeronautics and Astronautics, Paper AIAA, 2002–1274 (2002)
51. Petosa, A.: Dielectric Resonator Antenna Handbook. Artech House, Boston (2007)
52. RO4000 series high frequency circuit materials, data sheet. Publication no. 92-004, Rogers Corporation, Chandler, AZ (2010)
53. El Sabbagh, M.A., Bakr, M.H., Nikolova, N.K.: Sensitivity analysis of the scattering parameters of microwave filters using the adjoint network method. Int. J. RF Microw. Comput.-Aided Eng. **16**, 596–606 (2006)
54. Koziel, S., Ogurtsov, S., Bandler, J.W., Cheng, Q.S.: Robust space mapping optimization exploiting EM-based models with adjoint sensitivity. In: IEEE MTT-S Int. Microwave Symp. Dig. (2012)
55. Alexandrov, N.M., Lewis, R.M.: An overview of first-order model management for engineering optimization. Optim. Eng. **2**, 413–430 (2001)
56. Dielectric resonator filter, Examples, CST Microwave Studio, ver. 2011. CST AG, Bad Nauheimer Str. 19, D-64289 Darmstadt, Germany (2011)

Surrogate-Based Circuit Design Centering

Abdel-Karim S.O. Hassan and Ahmed S.A. Mohamed

Abstract Circuit design centering is one of the most important problems concerning the optimal design of circuits. Circuit design centering seeks nominal values of designable circuit parameters that maximize the probability of satisfying the design specifications (yield function). Design centering can be performed geometrically by finding the center of the feasible region (region in the designable parameter space where the design specifications are satisfied), or by maximizing the yield function explicitly. For all cases, the high expense of circuit simulations required obstructs the design centering process, especially for microwave circuits. To overcome this, computationally cheap surrogate-based models (e.g., space mapping, response surfaces, kriging, and neural networks) can be used for approximating the response functions or the yield function itself. In this chapter the design centering problem is formulated as an optimization problem, and the estimation of the yield function through several sampling techniques is explained. The difficulties facing the design centering process, especially for microwave circuits, are discussed, and the role of surrogate-based models in overcoming these difficulties is demonstrated. Special interest is devoted to space mapping surrogates and microwave circuit design centering. Some of the important surrogate-based circuit design centering approaches are reviewed with an overview of their theoretical bases. Tutorial and practical circuit examples are given to show the effectiveness of these approaches.

Keywords CAD · Design centering · Yield optimization · Surrogate-based optimization · Ellipsoidal technique · Semi-definite programming (SDP) · Trust region optimization · Space-mapping surrogates · EM-based design

A.-K.S.O. Hassan (✉) · A.S.A. Mohamed
Engineering Mathematics and Physics Department, Faculty of Engineering, Cairo University, Giza 12211, Egypt
e-mail: asho_hassan@yahoo.com

A.S.A. Mohamed
e-mail: aashiry@ieee.org

S. Koziel, L. Leifsson (eds.), *Surrogate-Based Modeling and Optimization*,
DOI 10.1007/978-1-4614-7551-4_2,

1 Introduction

In circuit design, circuits are characterized by some designable parameters $\mathbf{x} \in \mathbb{R}^n$ and circuit performance measures $f_i(\mathbf{x})$, $i = 1, 2, \ldots, m$. Designable circuit parameters may be passive elements or transistor geometries. The performance measures may be power dissipation or the circuit S-parameters. Naturally, circuit performance measures are functions of circuit responses which are evaluated through circuit simulations. The desired performance of a circuit (the design specifications) is described by specifying bounds on the performance measures of the circuit which is set by the designer. These design specifications constrain the designable parameters and define a region in the designable parameter space known as the feasible region (design region), which can be defined as:

$$F = \left\{ \mathbf{x} \in \mathbb{R}^n \mid f_i\big(\mathbf{R}(\mathbf{x})\big) \leq b_i,\ i = 1, 2, \ldots, m \right\}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the vector of the designable parameters, $\mathbf{R} : \mathbb{R}^n \to \mathbb{R}^m$ is the response vector, n is the number of designable parameters, m is the number of constraints, f_i is the i-th performance function, and b_i is the corresponding specification bound. Every $\mathbf{x} \in F$ is considered as an acceptable circuit.

In general, the traditional circuit design trend uses optimization techniques to find a nominal design of the circuit, i.e., to determine the nominal values of circuit parameters which satisfy the design specifications. It is a fact that circuit parameters are subject to known but unavoidable statistical fluctuations inherent to the manufacturing process used, due to environmental effects during operations, or due to model uncertainties. This may cause the circuit performance to violate the design specifications, especially when the location of the nominal design point in the designable parameter space is closer to the boundaries of the feasible region. To simulate the statistical fluctuations, circuit designable parameters are assumed to be random variables with a joint probability density function (PDF) $P(\mathbf{x}, \mathbf{x}_0)$, where $\mathbf{x}_0 \in \mathbb{R}^n$ is the vector of nominal parameter values. Therefore, the probability of satisfying the design specifications (yield function) can be defined as:

$$Y(\mathbf{x}_0) = \int_F P(\mathbf{x}, \mathbf{x}_0)\, dx. \tag{2}$$

The design centering problem seeks nominal values of circuit parameters which minimize the undesirable effects of the statistical fluctuations that affect the designable circuit parameters; namely, it seeks the nominal values of circuit parameters which maximize the yield function. Hence, the design centering or yield maximization problem is formulated as:

$$\max_{\mathbf{x}_0} \quad Y(\mathbf{x}_0). \tag{3}$$

2 Yield Function Estimation and Sampling Techniques

It is clear that the yield integral (2) cannot be evaluated analytically, since it requires the evaluation of an n-dimensional integral over a "non-explicitly defined" region

[31]. Instead, it can be estimated. One of the famous methods used for estimating the yield integral is the Monte Carlo method [41]. To verify this, we define the acceptance index function, $\mathrm{I_a} : \mathbb{R}^n \to \mathbb{R}$, as:

$$\mathrm{I_a}\left(\mathbf{x}^j\right) = \begin{cases} 1 & \text{if } \mathbf{x}^j \in F, \\ 0 & \text{if } \mathbf{x}^j \notin F, \end{cases} \tag{4}$$

where F is the feasible region defined by (1). Then, the yield integral (2) can be rewritten as:

$$Y(\mathbf{x}_0) = \int_{\mathbb{R}^n} \mathrm{I_a}(\mathbf{x}) P(\mathbf{x}, \mathbf{x}_0)\, dx = E\left\{\mathrm{I_a}(\mathbf{x})\right\}, \tag{5}$$

where $E\{\cdot\}$ denotes expectation.

Hence, the yield value at a nominal parameter vector $\mathbf{x}_0$ can be estimated by generating a set of sample points $\mathbf{x}^j$, $j = 1, 2, \ldots, K$ in the designable parameter space using the PDF of designable parameters. The circuit is simulated for each sample point $\mathbf{x}^j$, and the acceptance index function is evaluated. Hence, the yield function at the nominal parameter vector $\mathbf{x}_0$ can be estimated as:

$$Y(\mathbf{x}_0) \approx \frac{1}{K} \sum_{j=1}^{K} \mathrm{I_a}\left(\mathbf{x}^j\right) = \frac{k}{K}, \tag{6}$$

where k is the number of sample points satisfying the design specifications; i.e., the percentage of acceptable circuits gives an estimate of the yield value at $\mathbf{x}_0$. The error in estimating a yield value (the estimation variance) is given by the following formula [34]:

$$V\left(Y(\mathbf{x}_0)\right) \approx \frac{Y(\mathbf{x}_0)(1 - Y(\mathbf{x}_0))}{K - 1}. \tag{7}$$

It is obvious that the estimation variance is inversely proportional to the number of samples considered. Hence, the most straightforward way of improving accuracy and obtaining a low variance estimator is to increase the number of circuit simulations. This can result in prohibitively large requirements for computing time. Furthermore, there is no assurance that all regions of the parameter space will be explored equally well. Several techniques have been proposed for improving the accuracy of a Monte Carlo yield estimate without increasing the number of samples. These methods are called variance reduction techniques [25, 34].

2.1 Variance Reduction Techniques

The main objective of variance reduction techniques is to spread the sample data points as evenly as possible around the interior design space. The benefit is a fewer number of runs, to achieve the same level of confidence, than the number required by the Monte Carlo approach, because it is guaranteed that the entire probability range will be explored.

2.1.1 Importance Sampling

Importance sampling [34] uses another sampling density function in generating the sample points. To compensate this, the values of the acceptance index for sample points are multiplied by weighting factors equal to the ratio between the values of the original PDF and the sampling density function at the sample points. The proper choice of the sampling density function can lead to a variance smaller than that of the standard Monte Carlo yield estimator for the same sample size.

2.1.2 Stratified Monte Carlo Method

The most powerful feature of the Monte Carlo method is the fact that it is the only feasible and reliable method in large dimensions, since the number of required samples does not depend on the number of circuit parameters. The main drawback of the basic Monte Carlo sampling design is that the generated samples may leave large regions of the design region unexplored, as shown in Fig. 1(a) in a 2D case. A modified method to solve this deficiency is the stratified Monte Carlo method, in which stratified sampling is applied [34, 36]. In the stratified Monte Carlo method, each design parameter range is divided into subintervals (bins) of equal probability. A sampling site is then selected within each bin. Figure 1(b) shows an example in the 2D case with a uniform distribution of design variables (four bins for each design variable).

2.1.3 Latin Hypercube Sampling (LHS)

Latin hypercube sampling was developed in the work of McKay et al. [40]. It provides a more accurate estimate of the function value than the Monte Carlo method. The LHS involves dividing the design space into equiprobable subregions. Then N samples are selected such that all subregions are sampled.

One common way to generate N samples using the LHS design is to divide each design variable range into N nonoverlapping equiprobable intervals, each with a probability of $1/N$. Then N different values are selected for each variable at random (one for each interval). This process divides the n-dimensional design space into N^n cells with a probability $1/N^n$ each. The final N samples are obtained by selecting random combinations from the N values of all design variables. Figure 1(c) shows an example of LHS sampling for $n = 2$ design variables and $N = 4$ samples.

3 Circuit Design Centering

Circuit design centering is one of the most important problems concerning the optimal design of circuits [10, 26]. The design centering problem seeks nominal values of circuit parameters which maximize the yield function. Design centering approaches can be classified as statistical approaches and geometrical approaches. Statistical approaches optimize the yield function explicitly using statistical analysis

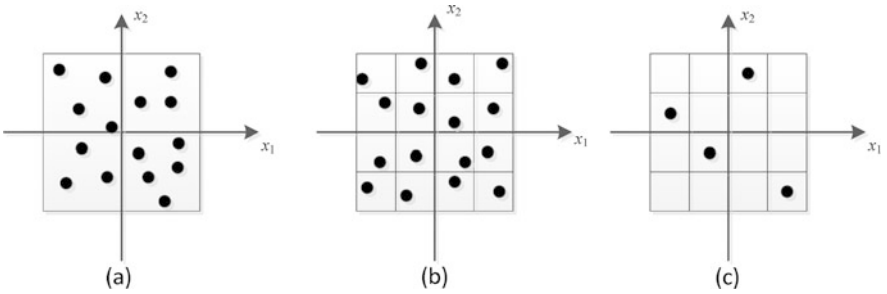


Fig. 1 (**a**) Basic Monte Carlo sampling. (**b**) Stratified Monte Carlo sampling. (**c**) Latin hypercube sampling for two dimensions with $N = 4$

techniques in a straightforward way regardless of the size of the problem or its convexity [23, 31, 32, 35, 36, 49, 53, 59, 60]. As previously stated, the accuracy of the statistical estimation does not depend on the number of parameters and performance features, but on the size of the sample [26]. An acceptable accuracy requires a large number of circuit simulations, in the range of thousands. Hence, the large computational effort required in addition to the uncertainty involved in the estimation process represent an obstacle against the statistical design centering approaches. Geometrical approaches, on the other hand, optimize the yield function implicitly by finding the center of the feasible region. This may be performed by approximating the feasible region using a convex body, e.g., a hyperellipsoid. Then the center of this body is considered as the design center [1–5, 7, 9, 22, 27–30, 33, 46, 48, 57, 58]. The geometrical approaches have fast convergence for convex and small-dimensional problems. Hybrid methods, which combine both approaches, may be used to overcome such problems [33].

4 Surrogate-Based Circuit Design Centering

In general, treating the feasible region during implemention of the geometrical design centering approaches requires evaluating the performance functions and perhaps also evaluating their gradients. Consequently, many circuit simulations will be needed during the design centering process. On the other hand, the statistical design centering process has some permanent special difficulties. For example, the cost of finding a yield value for a given nominal design parameter represents an obstacle for many optimization methods. Hence, robust optimization methods that utilize the fewest possible number of yield evaluations are required. Another difficulty is the absence of any gradient information, as the required simulations cost in evaluating the gradient information is prohibitive in practice [26]. Attempting to approximate the yield gradients using the finite difference approach requires many more yield evaluations, which highly increase the computational cost. Another objection in estimating the gradients by finite differencing is that the estimated yield values are usually contaminated by some numerical noise due to estimation uncertainty. In

these cases, for any small perturbations around a given point, the corresponding yield values do not reflect the local behavior of the yield function itself, but rather that of the noise [31]. Hence, gradient-based optimization methods cannot be applied here.

For all cases of design centering, the high expense of the required circuit simulations may obstruct the design centering process. One of the intelligent methods to overcome this obstacle is to use computationally cheap surrogate-based models. The surrogate-based models may be, for example, space mapping, response surfaces, kriging models, or neural networks. These surrogate-based models can be used for approximating the response functions and the yield function itself. The surrogate-based models are initially constructed and iteratively updated during the design centering process.

4.1 Surrogate-Based Statistical Design Centering Using Trust Region Optimization and Variance Reduction

A surrogate-based statistical design centering algorithm was introduced by Hassan et al. [31] to overcome the difficulties of statistical yield optimization. The algorithm neither requires nor approximates the gradients of the yield and the performance functions. It consists of two parts: a non-derivative unconstrained optimizer and a variance reduction estimator. The first part of the algorithm implements a non-derivative optimization method that combines a trust region framework with quadratic interpolating surrogates for the yield function [18, 19, 43]. The principal operation of the method relies on building, successively updating, and optimizing a quadratic surrogate of the yield function over trust regions. The quadratic surrogate of the yield function reasonably reflects the local behavior of the yield function in a trust region around a current iterate. A new point is then found by maximizing the surrogate model over the trust region. The second part of the algorithm utilizes the stratified Monte Carlo sampling technique [34, 36] for yield estimation during the optimization process. With this sampling technique, a lower variance yield estimator can be obtained which decreases the number of circuit simulations required to achieve a desired accuracy level.

In the given algorithm, the computationally expensive yield function is locally approximated around a current design point $\mathbf{x}_k$ by a computationally cheaper quadratic surrogate model $M(\mathbf{x})$, which can be placed in the form:

$$M(\mathbf{x}) = c + \mathbf{g}^T(\mathbf{x} - \mathbf{x}_k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^T \mathbf{B}(\mathbf{x} - \mathbf{x}_k), \tag{8}$$

where $c \in \mathbb{R}$, $\mathbf{g} \in \mathbb{R}^n$, and the symmetric matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$ are the unknown parameters of $M(\mathbf{x})$. The total number of these unknowns is $N = \frac{1}{2}(n+1)(n+2)$. These parameters are determined by interpolating the yield function at N interpolation points $\mathbf{x}_i$ using the matching conditions:

$$M(\mathbf{x}_i) = Y(\mathbf{x}_i), \quad i = 1, 2, \ldots, N. \tag{9}$$

However, for $n > 1$, the existence and uniqueness of $M(\mathbf{x})$ are not guaranteed by the number of interpolation points only. Some geometric restrictions on their positions must be satisfied [20, 47]. In other words, if expression (9) is written as a system of linear equations in the unknown parameters of $M(\mathbf{x})$, then the $N \times N$ coefficient matrix of this system (Vandermonde matrix) should be nonsingular for the quadratic interpolating model to be considered as a good approximation of the yield function around $\mathbf{x}_k$. On the other hand, typical distances between the interpolation points are taken to be of the magnitude of a positive scalar denoted ρ [43], which helps to reduce the contribution from the noise to the computations.

Assume that the current design point $\mathbf{x}_k$ in Eq. (8) is chosen to be the interpolation point that provides the best yield value so far, i.e.,

$$Y(\mathbf{x}_k) \geq Y(\mathbf{x}_i), \quad i = 1, 2, \ldots, N. \tag{10}$$

The model $M(\mathbf{s})$ is then maximized, in place of the yield function, over the current trust region and a new point is produced by solving the trust region subproblem:

$$\max_{\mathbf{s} \in \mathbb{R}^n} \quad M(\mathbf{s}) \quad \text{s.t.} \quad \|\mathbf{s}\| \leq \Delta, \tag{11}$$

where $\mathbf{s} = \mathbf{x} - \mathbf{x}_k$, $\Delta \geq \rho$ is the current trust region radius, and $\|\cdot\|$ throughout is the l_2-norm. This problem can be solved by the method of Moré and Sorensen [42]. Let $\mathbf{s}^*$ denote the solution of (11), and then a new point $\mathbf{x}_n = \mathbf{x}_k + \mathbf{s}^*$ is obtained. The ratio between the actual yield increase and the model increase obtained at this point is given by:

$$r = \frac{Y(\mathbf{x}_n) - Y(\mathbf{x}_k)}{M(\mathbf{x}_n) - M(\mathbf{x}_k)}. \tag{12}$$

This ratio reflects how much the surrogate model agrees with the yield function within the trust region. If there is good agreement, i.e., $r \geq 0.7$, the trust region radius Δ is enlarged. However, if the agreement is poor, $r < 0.1$, then Δ is reduced. For moderate r, i.e., $0.1 \leq r < 0.7$, Δ is considered suitable.

The newly obtained information represented in the pair $(\mathbf{x}_n, Y(\mathbf{x}_n))$ is exploited in modifying and improving the quadratic model, if possible. For $Y(\mathbf{x}_n) > Y(\mathbf{x}_k)$, one of the current interpolation points will be dropped and $\mathbf{x}_n$ will be included instead. In this situation, $\mathbf{x}_n$ enters the interpolation point set and a new iteration will then begin by maximizing the improved model over the new trust region. When the point $\mathbf{x}_n$ fails to increase the yield or modify the model, a procedure aims at checking and (possibly) improving the validity of the model around the current point $\mathbf{x}_k$. This procedure eliminates the worst interpolation point, $\mathbf{x}_j$ say, and searches for a new replacing point $\hat{\mathbf{x}}_j$ in the neighborhood $\|\mathbf{x} - \mathbf{x}_k\| \leq \rho$ [43]. The calculations will continue with the current value of ρ if and only if $\|\mathbf{s}^*\| > \rho$, where $\mathbf{s}^*$ is the solution of (11). Otherwise, no more iteration is required for the current value of ρ, since the model is considered valid but it seems that steps of length ρ fail to increase the yield. Hence, ρ is reduced, or termination occurs if either ρ reaches a final value or a prescribed yield increase is achieved.

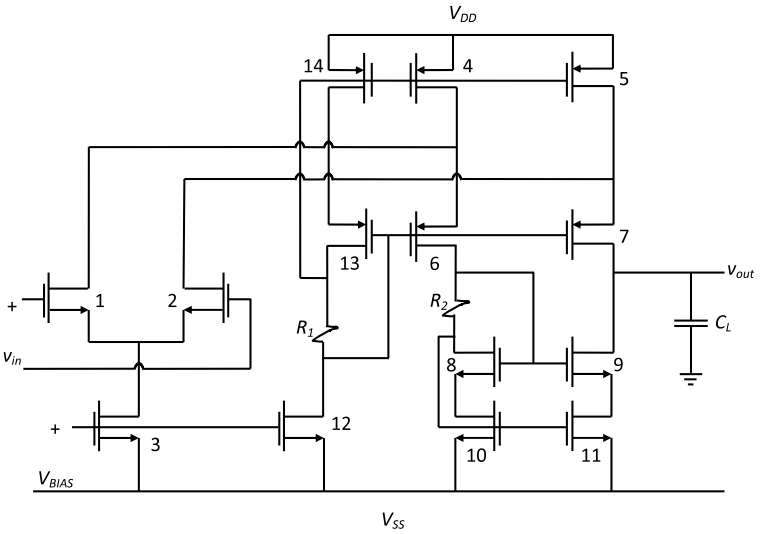


Fig. 2 A folded cascode operational amplifier [6]

Table 1 Performance specifications for the folded cascode operational amplifier [31]

Performance measure	Specification	Initial value	Final value (indep. par)	Final value (correlated parameters)
Slew rate	$\geq$10 V/µs	61	18.07	23.215
Power dissipation	$\leq$10 mW	3.4	0.619	1.0125
Gain bandwidth product	$\geq$10 MHz	49.3	25.6	26.7
Maximum output voltage	$\geq$2 V	1.84	2.177	2.07
Minimum output voltage	≤ -2 V	−1.98	−2.216	−2.157
Maximum input CM voltage	$\geq$2.5 V	2.87	3.028	2.98
Minimum input CM voltage	≤ -1.3 V	−1.28	−1.52	−1.4
Differential voltage gain	$\geq$5,000	4,009	14,266	10,353
Area of designable transistors	$\leq$300 (µm)^2	176	245	260
Minimum channel length	$\geq$0.8 µm	0.8	1.1	0.966
Minimum channel width	$\geq$10 µm	30	21.3	15.05

4.2 Folded Cascode Operational Amplifier Design Centering

The optimal design of a folded cascode operational amplifier [6] is performed using the given algorithm. The amplifier is shown in Fig. 2. This is a 12D problem in which the designable parameters are the widths and lengths of transistors M_1, M_3, M_4, M_6, and M_8 (in micrometers), together with the load capacitor C_L (in picofarads) and the biasing voltage V_{bias} (in volts). There are 11 performance mea-

sures of interest given in Table 1. In this table, the area is defined as the sum of the product of the widths and lengths of all the designable transistors. Table 1 also shows the specification bounds of the performance measures, and their values at the initial and final design parameters. The parameter variance vector is assumed to be $[0.125\ 0.125\ 0.125\ 0.125\ 0.125\ 0.5\ 0.5\ 0.5\ 0.5\ 0.5\ 0.15\ 0.125]^T$ and the maximum cross-correlation coefficient is 0.8. The algorithm starts from a bad initial design $[0.8\ 0.8\ 0.8\ 0.8\ 0.8\ 30.0\ 30.0\ 70.0\ 60.0\ 30.0\ 3.0\ -1.5]^T$ at which the yield is 0.2 %. The algorithm reaches a final design for independent parameters $[1.23\ 1.19\ 1.09\ 1.25\ 1.14\ 29.22\ 25.59\ 69.12\ 59.75\ 5.47\ 1.83\ -1.632]^T$ at which the yield is 92.6 % by using 246 yield evaluations. For the case of correlated parameters, the algorithm reaches the final design center $[1.19\ 1.57\ 1.16\ 1.04\ 1.2\ 30.59\ 23.77\ 70.37\ 65.85\ 30.19\ 2.33\ -2.06]^T$ at which the yield is 93 % by using 259 yield evaluations. A comparison between the stratified sampling and the Monte Carlo method showed that the standard deviation of the stratified sampling yield estimator by using 1,000 samples is almost upper bounded by that of the Monte Carlo estimator by using 1,800 samples, which saves about 44 % of the circuit simulations [31].

The effectiveness of the given algorithm is tested by applying another optimizer for yield maximization. The optimizer uses the Nelder and Mead simplex method with quadratic response surface [24]. After an upper bound of 1,000 yield evaluations, yields of only 12.9 % for independent parameters and 23.5 % for the correlated parameters are obtained.

5 Surrogate-Based Microwave Circuit Design Centering

Design centering of nonlinear microwave circuits is a great challenge [12]. The computational overhead is one of the main difficulties in the design centering process of these circuits, as many full-wave electromagnetic (EM) simulations would be required. For both statistical and geometrical design centering approaches, the high cost of expensive EM circuit simulations required obstructs the design centering process. To overcome this, computationally cheap surrogate-based models can be used for approximating the response functions and the yield function itself.

The problem of high computational cost required for microwave design centering has been solved successfully by using space mapping interpolating surrogates (SMISs) [5, 32]. Space mapping (SM) techniques employ computationally fast, coarse models to greatly reduce the evaluation cost of the computationally expensive full-wave EM fine models [8, 13–15, 17, 37]. The SMIS technique aims to calibrate a space-mapped surrogate, via input and output mappings, to match the fine model with high accuracy.

As previously given, the desired performance of a microwave circuit is described by some performance specifications which define the feasible region in the designable parameter space:

$$F = \left\{ \mathbf{x} \in \mathbb{R}^n \ \mid \ f_i\left(\mathbf{R}_f(\mathbf{x})\right) \ \leq b_i,\ i = 1, 2, \ldots, m \right\}, \tag{13}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the vector of the designable parameters, $\mathbf{R}_f : \mathbb{R}^n \to \mathbb{R}^m$ is the fine model response vector, n is the number of design parameters, m is the number of constraints, f_i is the i-th performance function, and b_i is the corresponding specification bound. However, working with (13) involves a lot of computationally expensive fine model evaluations. Instead, SMISs can be employed, and the feasible region is approximated by:

$$F_s = \left\{ \mathbf{x} \in \mathbb{R}^n \mid f_i\left(\mathbf{R}_s(\mathbf{x})\right) \leq b_i,\ i = 1, 2, \ldots, m \right\}, \tag{14}$$

where F_s is the SM feasible region approximation, and $\mathbf{R}_s : \mathbb{R}^n \to \mathbb{R}^m$ is the SMIS response vector. Two matching conditions have to be satisfied:

$$\mathbf{R}_s(\mathbf{x}) = \mathbf{R}_f(\mathbf{x}), \tag{15}$$

$$\mathbf{J}_s(\mathbf{x}) = \mathbf{J}_f(\mathbf{x}), \tag{16}$$

where $\mathbf{J}_s$ and $\mathbf{J}_f$ are the Jacobian matrices of the surrogate and fine model, respectively. These matrices contain the first-order partial derivatives of the response vector with respect to designable parameters.

5.1 Generalized Space Mapping (GSM) Surrogate Model

The GSM surrogate model is constructed, using a computationally fast coarse model with input and output mappings, in the form [37]:

$$\mathbf{R}_s^i(\mathbf{x}) = \mathbf{A}^i.\mathbf{R}_c\left(\mathbf{B}^i.\mathbf{x} + \mathbf{c}^i\right) + \mathbf{d}^i + \mathbf{E}^i\left(\mathbf{x} - \mathbf{x}_0^i\right), \tag{17}$$

where $\mathbf{x}_0^i$ is the current nominal parameter vector, $\mathbf{A}^i \in M_{m\times m}$ is a diagonal matrix, $\mathbf{R}_c : X_c \to \mathbb{R}^m$ is the coarse model response vector, $\mathbf{B}^i \in M_{n\times n}$, $\mathbf{c}^i \in M_{n\times 1}$, and $\mathbf{d}^i \in M_{m\times 1}$ is given:

$$\mathbf{d}^i = \mathbf{R}_f\left(\mathbf{x}_0^i\right) - \mathbf{A}^i \cdot \mathbf{R}_c\left(\mathbf{B}^i.\mathbf{x}_0^i + \mathbf{c}^i\right), \tag{18}$$

where $\mathbf{R}_f : X_f \to \mathbb{R}^m$ is the fine model response vector, and $\mathbf{E}^i \in \mathrm{M}_{m\times n}$ is

$$\mathbf{E}^i = \mathbf{J}_f\left(\mathbf{x}_0^i\right) - \mathbf{A}^i \cdot \mathbf{J}_c\left(\mathbf{B}^i.\mathbf{x}_0^i + \mathbf{c}^i\right).\mathbf{B}^i, \tag{19}$$

where $\mathbf{J}_f : X_f \to \mathbb{R}^{m\times n}$ and $\mathbf{J}_c : X_c \to \mathbb{R}^{m\times n}$ are the Jacobian matrices of the fine and coarse model responses with respect to the corresponding points, respectively. The mapping parameters $\mathbf{A}^i, \mathbf{B}^i, \mathbf{c}^i$ are obtained by the parameter extraction (PE) optimization process given by:

$$\left(\mathbf{A}^i, \mathbf{B}^i, \mathbf{c}^i\right) = \arg\min_{\mathbf{A},\mathbf{B},\mathbf{c}} \mathbf{e}^i(\mathbf{A}, \mathbf{B}, \mathbf{c}), \tag{20}$$

where $\mathbf{e}^i$ represents the response deviation residual of the surrogate from the fine model and is given by:

$$\mathbf{e}^i(\mathbf{A},\mathbf{B},\mathbf{c}) = \sum_{k=0}^{i} w_k \left\| \mathbf{R}_f\left(\mathbf{x}^k\right) - \mathbf{A}\cdot\mathbf{R}_c\left(\mathbf{B}.\mathbf{x}^k + \mathbf{c}\right)\right\| + \sum_{k=0}^{i} v_k \left\| \mathbf{J}_f\left(\mathbf{x}^k\right) - \mathbf{A}\cdot\mathbf{J}_c\left(\mathbf{B}.\mathbf{x}^k + \mathbf{c}\right).\mathbf{B}\right\|, \quad (21)$$

where the coefficients w_k and v_k are chosen according to the nature of the design problem.

5.2 *The Ellipsoidal Technique for Design Centering of Microwave Circuits*

In this work, a SM technique is integrated with an ellipsoidal technique to obtain a surrogate-based geometrical design centering method of microwave circuits with a small number of EM simulations [5]. The ellipsoidal technique [2, 3] approximates the feasible region with a hyperellipsoid which is the final hyperellipsoid of a generated sequence of decreasing volume of different shape and center hyperellipsoids. The center of this final hyperellipsoid is considered as a design center. The generation of the sequence of hyperellipsoids requires successive linearization of the feasible region boundaries at selected boundary points. This requires evaluating the performance functions and their gradients. Consequently, many expensive circuit simulations will be needed, especially for microwave circuits.

In this design centering method, an SMIS [14] is initially constructed and then updated through SM iterations. In each SM iteration, a current SMIS model and the corresponding SM feasible region approximation (14) are available. The ellipsoidal technique is implemented with the current feasible region approximation to obtain a new design centering point. This new center is validated by the fine model and is used to update the current SMIS model. Enhanced improvement is achieved by satisfying (15) at all preceding design centering points (global matching). The ellipsoidal technique is then restarted with the updated SMIS to get the next center. The process is repeated until a final design center is obtained.

5.2.1 The Ellipsoidal Technique

Assume that $\mathbf{x}_0^k$ is the design center obtained in the $(k-1)$-th SM iteration. In the k-th iteration, the feasible region approximation is $F_s^k = \{\mathbf{x} \in \mathbb{R}^n \mid f_i(\mathbf{R}_s^k(\mathbf{x})) \leq b_i, i = 1, 2, \ldots, m\}$. The ellipsoidal technique starts with a sufficiently large hyperellipsoid containing the feasible region approximation with an initial center $\mathbf{t}_0 = \mathbf{x}_0^k$. An iteration of the ellipsoidal technique assumes that the current hyperellipsoid is $\mathbf{E}_j = \{(\mathbf{x}-\mathbf{t}_j)^T \mathbf{Q}_j^{-1}(\mathbf{x}-\mathbf{t}_j) \leq 1\}$ with center $\mathbf{t}_j$ and hyperellipsoid matrix $\mathbf{Q}_j$ which is symmetric and positive definite. A hyperplane $\mathbf{a}^T\mathbf{x} = b$ is then constructed by linearizing the feasible region boundary at a selected boundary point. This hyperplane

divides the current hyperellipsoid into two parts. The first is completely infeasible, while the second contains the feasible region approximation. The new generated hyperellipsoid $\mathbf{E}_{j+1}(\mathbf{t}_{j+1}, \mathbf{Q}_{j+1})$ is the minimum volume hyperellipsoid enclosing the second part with center and matrix given by:

$$\mathbf{t}_{j+1} = \mathbf{t}_j - \eta \frac{(\mathbf{Q}_j \mathbf{a})}{\sqrt{\mathbf{a}^T \mathbf{Q}_j \mathbf{a}}}, \qquad \mathbf{Q}_{j+1} = \beta \left(\mathbf{Q}_j - \gamma \frac{(\mathbf{Q}_j \mathbf{a})(\mathbf{Q}_j \mathbf{a})^T}{\mathbf{a}^T \mathbf{Q}_j \mathbf{a}} \right), \tag{22}$$

where

$$\begin{aligned} &\eta = (1 + n\delta)/(n+1), \qquad \beta = n^2\big(1 - \delta^2\big)/\big(n^2 - 1\big), \\ &\gamma = 2\eta/(1+\delta) \quad \text{and} \quad \delta = \big(\mathbf{a}^T \mathbf{t}_j - b\big)/\sqrt{\mathbf{a}^T \mathbf{Q}_j \mathbf{a}}. \end{aligned} \tag{23}$$

The volumes of $\mathbf{E}_{j+1}$ and $\mathbf{E}_j$ satisfy the ratio [2]:

$$\rho = \rho_0 \big(1 - \delta^2\big)^{\frac{(n-1)}{2}} (1 - \delta), \quad \rho_0 = \frac{(n^n)}{((n+1)^{\frac{(n+1)}{2}} (n-1)^{\frac{(n-1)}{2}})}. \tag{24}$$

The hyperellipsoid volume ratio ρ is a monotonically decreasing function in δ. Thus, different strategies are used to locate the best boundary points, which are accompanied by maximum values of δ. This results in the greatest reduction in the volume of the generated hyperellipsoids and consequently increases the convergence of the ellipsoidal technique [3].

The iterations of the ellipsoidal technique continue until no significant reduction in the hyperellipsoid volume can be achieved. The center of the final hyperellipsoid, denoted by $\mathbf{t}_{\text{final}}$, considers the next design center $\mathbf{x}_0^{k+1}$ and is fed into the next SM iteration [5].

Note that the Broyden formula [16] offers a fast way to approximate the gradients required in linearization. However, for some models, exact gradients can be evaluated by the adjoint sensitivity technique [11, 50].

5.2.2 Design Centering of Coupled-Line Bandpass Filter

The optimal design of a coupled-line bandpass filter [52], shown in Fig. 3, is determined. The design constraint functions are given by:

$$f_i\big(\mathbf{R}_f(\mathbf{x})\big) = \begin{cases} R_{f,i}(\mathbf{x}) + 20, & 5 \text{ GHz} < \omega_i < 7.25 \text{ GHz} \\ -3 - R_{f,i}(\mathbf{x}), & 7.75 \text{ GHz} < \omega_i < 8.25 \text{ GHz} \\ R_{f,i}(\mathbf{x}) + 20, & 8.75 \text{ GHz} < \omega_i < 11 \text{ GHz} \end{cases}$$

where $R_{f,i}(\mathbf{x}) = |S_{21}|$ (dB) at frequency ω_i. The substrate thickness is taken as 1.272 mm and $\varepsilon_r = 10$. The design parameters are $[\,x_1\ x_2\ x_3\ x_4\ x_5\ x_6\,]^T$ in millimeters, as shown in Fig. 3. The simulation of this example is performed using an in-house planar solver based on the method of moments [51]. The fine model is meshed with 3 width and 15 length segments of the microstrip lines, while the coarse model is meshed with 1 width and 5 length segments.

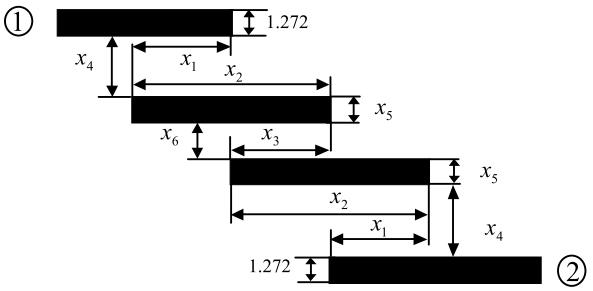


Fig. 3 Coupled-line bandpass filter [52]

Table 2 Yield results for the coupled line bandpass filter assuming independent parameters [5]

Parameter spreads σ	Initial yield	Yield at minimax solution	Final yield
$\sigma_i = 2\ \%$[a]	2 %	5 %	14 %
$\sigma_i = 1\ \%$[a]	0 %	20 %	34 %
$\sigma_i = 0.5\ \%$[a]	0 %	37 %	55 %
$\sigma_i = 0.1019$[b]	0 %	6 %	17 %
$\sigma_i = 0.034$[b]	0 %	45 %	75 %

[a]Parameter spreads are relative to the nominal values at the final solution

[b]Absolute parameter spreads in millimeters

Table 3 Yield results for the coupled-line bandpass filter assuming correlated parameters [5]

Covariance matrix	Initial yield	Yield at minimax solution	Final yield
$\mathbf{Cov}_1$	0 %	10 %	27 %
$\mathbf{Cov}_2$	0 %	57 %	93 %

The constraint functions are evaluated at all frequency points $\omega_i = 5$, 5.75, 6.5, 7.25, 7.75, 8, 8.25, 8.75, 9.5, 10.25, 11 GHz. The initial surrogate is taken as the coarse model giving $[5.682\ 13.991\ 6.835\ 0.522\ 1.051\ 1.511]^T$ as the initial fine model design. The final design center $[5.298\ 12.960\ 6.052\ 0.416\ 2.122\ 1.099]^T$ is reached after six SM iterations. The initial and the final yield values are evaluated via the Monte Carlo method with 100 sample points assuming normally distributed parameters. The results assuming independent parameters are shown in Table 2.

The results for the correlated parameters are shown in Table 3. A much higher yield is achieved with the obtained design center in comparison with the minimax center $[4.779\ 13.664\ 6.835\ 0.637\ 1.024\ 0.808]^T$. In Table 3, $\mathbf{Cov}_1$ and, $\mathbf{Cov}_2$ are final hyperellipsoid matrices scaled to give hyperellipsoids of the same volume as the hyperellipsoids with independent parameter spreads $\sigma_i = 0.1019$ and $\sigma_i = 0.034$.

5.3 Design Centering of Microwave Circuits via Trust Region Optimization and Space Mapping Surrogates

This approach of design centering is a statistical one and treats two types of surrogates [32]. It employs the GSM surrogate (17), to minimize the required expensive EM simulations, in addition to the quadratic surrogate model (8) to approximate the computationally expensive yield function. The approach integrates three strategies to overcome the statistical microwave design centering difficulties. First, Latin hypercube sampling (LHS) is used in the sampling process. Second, a derivative-free trust region method is utilized in the yield optimization process. In the given approach, the NEWUOA algorithm developed by Powell is used [44, 45, 61]. The NEWUOA algorithm employs quadratic surrogate models to approximate the expensive yield function. Third, the GSM technique [37] is employed to reduce the simulation computational effort. Thus, the NEWUOA algorithm is combined with the GSM technique and the LHS technique to obtain a method for statistical microwave circuit design centering.

Starting from an initial point, the GSM surrogate model is initially constructed, and then updated through SM iterations. In each SM iteration, a current SMIS model and the corresponding SM feasible region approximation are available. The NEWUOA algorithm is applied to optimize the yield function and get a better center point. The yield values are estimated using the current feasible region approximation. The new center point is validated by the fine model and is used to update the current SMIS using the matching conditions. Then, the NEWUOA algorithm starts again with the updated surrogate to obtain the next center point.

5.3.1 NEWUOA Algorithm

In the NEWUOA algorithm, the computationally expensive yield function is locally approximated around a current iterate utilizing the quadratic surrogate model $M(\mathbf{x})$ in (8) by interpolating the yield function at $m = 2n + 1$ points. The total number of unknown parameters in the surrogate quadratic model is $N = ½(n + 1)(n + 2)$. The freedom in $M(\mathbf{x})$ is taken up by minimizing the Frobenius norm of the change in the Hessian matrix $\mathbf{B}$ during the optimization process, i.e., $\|\mathbf{B}^{\text{new}} - \mathbf{B}^{\text{old}}\|_F$ [44]. The estimated yield function values are submitted to the optimizer by a subroutine that employs the LHS technique, and the generated samples are tested against the current feasible region approximation.

The quadratic model in (8) is then maximized, instead of the yield function, over a current trust region (TR) by solving the TR subproblem (11). The TR radius is revised according to the agreement between the quadratic model and the yield function at the new point. The TR radius Δ has a lower bound ρ in the interval $[\rho_{\text{fin}},\ \rho_{\text{ini}}]$. This parameter ρ is utilized to maintain enough distances between the interpolation points where ρ_{ini} and ρ_{fin} are user-defined initial and final radii, respectively. Let Δ_{old} and Δ_{new} be the old and new values of Δ. The choice of Δ_{new} depends on the ratio between the actual yield increase and the model increase as in (12), and

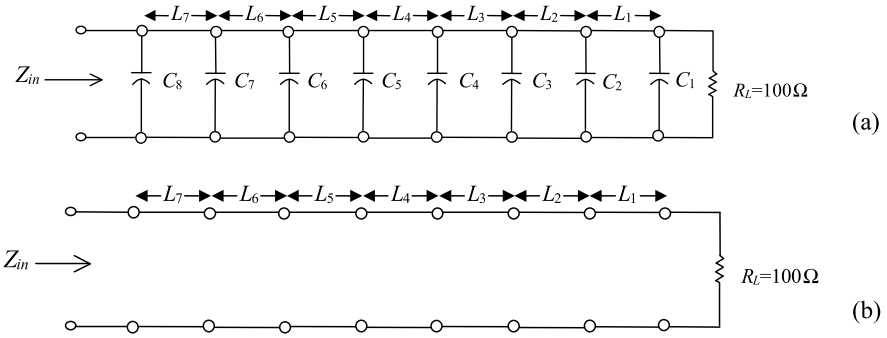


Fig. 4 Seven-section capacitively loaded impedance transformer: (**a**) fine model, (**b**) coarse model

the Euclidean length of the step **s** obtained when solving (11). NEWUOA sets Δ_{new} to ρ or to Δ_{int} when $\Delta_{\text{int}} < 1.5\rho$ or $\Delta_{\text{int}} > 1.5\rho$, respectively, where Δ_{int} is the intermediate value [45]:

$$\Delta_{\text{int}} = \begin{cases} \|\mathbf{s}\|, & r \le 0.1 \\ \max\{\|\mathbf{s}\|,\ 0.5\ \Delta_{\text{old}}\}, & 0.1 < r \le 0.7 \\ \max\{\|\mathbf{s}\|, \Delta_{\text{old}}\}, & r > 0.7 \end{cases} \tag{25}$$

The algorithm is terminated when the TR radius reaches the lower bound ρ_{fin} that fixes the final accuracy required in the parameters [44, 45]. Numerically, NEWUOA shows good results and acceptable accuracy in problems with dimensions up to 320 variables [45].

5.3.2 Design Centering of Seven-Section Transmission Line Capacitively Loaded Impedance Transformer

The seven-section transmission line (TL) capacitively loaded impedance transformer is described in [8]. The coarse model is considered as an ideal seven-section TL, where the "fine" model is a capacitively loaded TL with capacitors $C_1 = \cdots = C_8 = 0.025$ pF (see Fig. 4). The design parameters are $\mathbf{x} = [L_1\ L_2\ L_3\ L_4\ L_5\ L_6\ L_7]^T$, which are the normalized lengths with respect to the quarter-wavelength L_{q} at the center frequency 4.35 GHz. The design specifications are:

$$f_i\left(\mathbf{R}_f(\mathbf{x})\right) = \left|\mathbf{S}_{11}(\mathbf{x}, \omega_i)\right| \le 0.07, \quad 1\ \text{GHz} \le \omega_i \le 7.7\ \text{GHz}$$

with 68 points per frequency sweep.

An initial infeasible point $[0.892\ 0.993\ 0.989\ 0.981\ 0.996\ 0.99\ 0.891]^T$ is considered. The yield values are estimated via the LHS method with 200 sample points assuming normally distributed parameters with covariance matrices G/49 and G/441, where $[0.193\ 0.194\ 0.145\ 0.046\ 0.155\ 0.239\ 0.38]^T$ is a parameter variance vector of the covariance matrix G with maximum cross-correlation coefficient 0.15.

Table 4 Results of the seven-section TL transformer with normally distributed correlated parameters

Covariance matrix	Initial yield		Final yield	
	Surrogate	Fine	Surrogate	Fine
G/49	2.5 %	3.0 %	30.0 %	27.5 %
G/441	0.2 %	0.2 %	99.2 %	100 %

The technique needed three SM iterations to obtain the results shown in Table 4. The yield results at the initial and final designs are shown in Fig. 5.

5.4 Microwave Design Centering via Semidefinite Programming and Space Mapping Surrogates

The method has a statistical-geometrical nature [33]. It exploits semidefinite programming [21, 54–56] and GSM surrogates [37] to approximate the feasible region with two bounding ellipsoids. The centers of these ellipsoids are used as design centers. The bounding ellipsoids are obtained using a two-phase algorithm. In the first phase, the minimum volume ellipsoid enclosing the feasible region is obtained, while the largest ellipsoid that can be inscribed within the feasible region is obtained in the second phase.

5.4.1 Phase (I): Minimum Volume Ellipsoid Enclosing the Feasible Region

The first phase of the method starts with an initial point and an initial GSM surrogate model. This surrogate model is iteratively updated through SM iterations. In each SM iteration a current feasible region approximation is available and a convergent sequence of increasing-volume Löwner–John ellipsoids is generated [21]. The

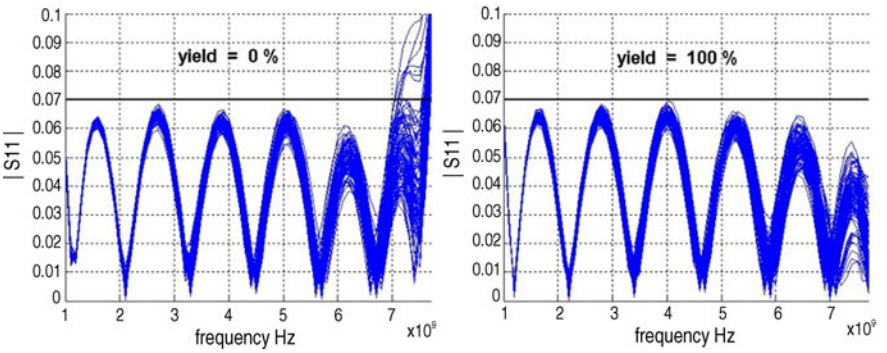


Fig. 5 Yield evaluated at the initial and final design parameter vectors with 100 sample points with covariance matrix G/441 for the seven-section capacitively loaded impedance transformer

generated ellipsoids enclose tightly selected sets of feasible points out of generated sample points. The generated sequence of Löwner–John ellipsoids converges to the minimum volume ellipsoid enclosing the current feasible region approximation, and its center is considered as a design center point. This new point is used to update the current GSM surrogate model using matching conditions. Consequently, a new updated feasible region approximation is constructed, and the process is repeated until the minimum volume ellipsoid is obtained.

Assume that $\mathbf{x}_0^k$ is the design center obtained in the $(k-1)$-th SM iteration. In the k-th iteration, the current feasible region approximation is $F_s^k = \{\mathbf{x} \in \mathbb{R}^n \mid f_i(\mathbf{R}_s^k(\mathbf{x})) \le b_i,\ i = 1, 2, \ldots, m\}$. In the k-th SM iteration, the initial point $\mathbf{q}_{\text{init}} = \mathbf{x}_0^k$ and an initial covariance matrix $\mathbf{B}_{\text{init}}$ are used to generate an LHS normally distributed sample set $\mathrm{S}^{(0)}$. Circuit simulation is executed on the sample set $\mathrm{S}^{(0)}$ in order to determine the feasible points and find the feasible set $\mathrm{S}_f^{(0)} = \mathrm{S}^{(0)} \cap F_s^k$. Then, the minimum volume ellipsoid $\mathcal{E}_{\text{Min}}^{(0)}(\mathbf{q}_{\text{min}}^{(0)}, \mathbf{B}_{\text{min}}^{(0)})$ enclosing the feasible set $\mathrm{S}_f^{(0)}$ (Löwner–John ellipsoid) is constructed as follows.

Let $\mathrm{S}_f^{(0)} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_p\} \subset \mathbb{R}^n$. Define the ellipsoid,

$$\mathcal{E}(\mathbf{q}, \mathbf{B}) = \left\{\mathbf{z} \in \mathbb{R}^n : (\mathbf{z}-\mathbf{q})^T \mathbf{B}^{-1} (\mathbf{z}-\mathbf{q}) \le \mathbf{1}\right\}, \tag{26}$$

where $\mathbf{q} \in \mathbb{R}^n$ is the center of the ellipsoid, and $\mathbf{B} \in \mathbf{SR}^n, \mathrm{B} \succ 0$ (symmetric and positive definite). Assume that $\mathbf{B} = \mathbf{Q}\mathbf{Q}^T$, where $\mathbf{Q}$ is nonsingular. Without loss of generality, it can be assumed that $\mathbf{Q} \in \mathbf{SR}^n, \mathbf{Q} \succ 0$ [28].

Hence, the minimum volume ellipsoid $\mathcal{E}_{\text{Min}}^{(0)}(\mathbf{q}_{\text{min}}^{(0)}, \mathbf{B}_{\text{min}}^{(0)})$ enclosing the set $S_f^{(0)}$ can be obtained by solving the following determinant maximization problem [54–56]:

$$\begin{aligned} &\max \quad \log\det \mathbf{A}_0 \\ &\text{subject to} \quad \begin{pmatrix} \mathbf{I}_n & \mathbf{A}_0\mathbf{z}_j + \mathbf{b}_0 \\ (\mathbf{A}_0\mathbf{z}_j + \mathbf{b}_0)^T & \mathbf{1} \end{pmatrix} \succcurlyeq \mathbf{0}, \quad j = 1, 2, \ldots, p, \\ &\qquad \mathbf{A}_0 = \mathbf{A}_0^T, \quad \mathbf{A}_0 \succ \mathbf{0}, \end{aligned} \tag{27}$$

where $\mathbf{I}_n$ is the n-dimensional identity matrix, $\mathbf{A}_0 = \mathbf{Q}_{\text{min}}^{(0)-\mathbf{1}}$, $\mathbf{b}_0 = -\mathbf{Q}_{\text{min}}^{(0)-\mathbf{1}}\mathbf{q}_{\text{min}}^{(0)}$,

This problem can be solved using semidefinite programming techniques [21, 54]. Clearly, obtaining the optimization variables $\mathbf{b}_0$ and $\mathbf{A}_0$ characterizes the Löwner–John ellipsoid $\mathcal{E}_{\text{Min}}^{(0)}(\mathbf{q}_{\text{min}}^{(0)}, \mathbf{B}_{\text{min}}^{(0)})$. The most distant feasible samples of $S_f^{(0)}$, which carry the dominant information about the ellipsoid $\mathcal{E}_{\text{Min}}^{(0)}$, are drawn into a new set S_{shell},

$$S_{\text{shell}} = \left\{\mathbf{z} \in S_f^{(0)} : r_{\text{shell}}^2 \le \left(\mathbf{z} - \mathbf{q}_{\text{min}}^{(0)}\right)^T \mathbf{B}_{\text{min}}^{(0)-\mathbf{1}} \left(\mathbf{z} - \mathbf{q}_{\text{min}}^{(0)}\right) \le 1\right\}, \tag{28}$$

where $0 \le r_{\text{shell}} \le 0.98$.

Using $\mathbf{q}_{\text{min}}^{(0)}$ and $\mathbf{B}_{\text{min}}^{(0)}$, a generation of LHS normally distributed sample set $S^{(1)}$ is performed. All the new samples of $S^{(1)}$ falling inside the current Löwner–John

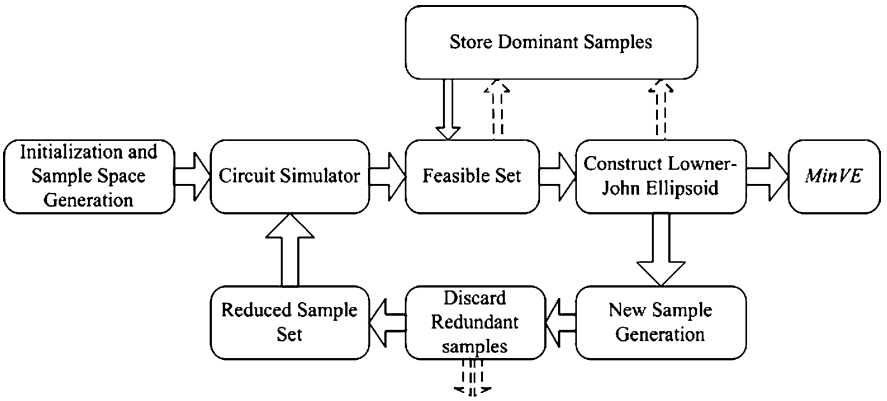


Fig. 6 Block diagram for SM iteration of phase (I) [33]

ellipsoid $\mathcal{E}_{\text{Min}}^{(0)}(\mathbf{q}_{\text{min}}^{(0)}, \mathbf{B}_{\text{min}}^{(0)})$, or a reduced version of it, are deleted, and a reduced sample set S_{reduced} is constructed for $0.9 \leq r_{\text{discard}} \leq 1$ as:

$$S_{\text{reduced}} = \left\{ \mathbf{z} \in \mathbf{S}^{(1)} : \left(\mathbf{z} - \mathbf{q}_{\text{min}}^{(0)}\right)^T \mathbf{B}_{\text{min}}^{(0)-1} \left(\mathbf{z} - \mathbf{q}_{\text{min}}^{(0)}\right) \geq r_{\text{discard}}^2 \right\}. \tag{29}$$

Circuit simulation is performed on the reduced sample set S_{reduced}, giving a feasible set $S_f = S_{\text{reduced}} \cap F_s^k$. The stored feasible samples in the set S_{shell} (28) are retrieved and added to the new feasible points of S_f, giving the first iteration's feasible set $S_f^{(1)} = S_f \cup S_{\text{shell}}$. Then, a new Löwner–John ellipsoid $\mathcal{E}_{\text{Min}}^{(1)}(\mathbf{q}_{\text{min}}^{(1)}, \mathbf{B}_{\text{min}}^{(1)})$ can be constructed using the feasible set $S_f^{(1)}$. Since the feasible set $S_f^{(1)}$ carries information about the last Löwner–John ellipsoid, an increased volume ellipsoid is attained, and the convergence of the method is guaranteed. Note that the reduced sample set S_{reduced} saves about 50–60 % of the required circuit simulations if the set $S^{(1)}$ was considered completely [33].

By repeating this process, a sequence of increasing-volume Löwner–John ellipsoids is generated to enclose the current feasible region approximation F_s^k. A suitable stopping criterion is chosen such that the minimum volume ellipsoid $\mathcal{E}_{\text{MinVE}}$ enclosing F_s^k is reached. The center of this ellipsoid is used to update the current GSM surrogate model using matching conditions. Consequently, a new updated feasible region approximation is constructed, and the process is repeated until the minimum volume ellipsoid $\mathcal{E}_{\text{MinVE}}$ enclosing the feasible region is obtained; its center is considered as a design center. Figure 6 shows an SM iteration of phase (I).

5.4.2 Phase (II): Maximum Volume Inscribed Ellipsoid

The second phase of the method treats the final SM feasible region approximation obtained in phase (I). It begins by constructing an initial polytope P_0 containing the minimum volume ellipsoid $\mathcal{E}_{\text{MinVE}}$. The initial polytope hyperplanes are chosen perpendicular to the ellipsoid axes, and passing through their end points. The

initial maximum volume ellipsoid $\mathcal{E}^{(0)}_{\text{Max}}(\mathbf{q}^{(0)}_{\text{max}}, \mathbf{B}^{(0)}_{\text{max}})$ inscribed in the polytope P_0 is the minimum volume ellipsoid $\mathcal{E}_{\text{MinVE}}$ containing the feasible region. An updated polytope P_1 is constructed by adding new hyperplanes obtained by linearizing the feasible region boundaries at selected boundary points. These points are obtained by searching along orthogonal directions, e.g., the parameter directions and/or the current ellipsoid axes, starting from the current ellipsoid's center.

An ellipsoid in $\mathbb{R}^n$ can be defined as [28]:

$$\mathcal{E}(\mathbf{q}, \mathbf{Q}) = \left\{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} = \mathbf{q} + \mathbf{Q}\mathbf{s} \text{ and } \|\mathbf{s}\| \leq 1\right\} \tag{30}$$

where $\mathbf{q} \in \mathbb{R}^n$ is the center, $\mathbf{Q} \in \text{SR}^n$. Assume that the polytope P_1 is given by:

$$\text{P}_1 = \left\{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b}\right\}, \tag{31}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m > n$, $\mathbf{b} \in \mathbb{R}^m$. Then, $\mathcal{E}^{(1)}_{\text{Max}}(\mathbf{q}^{(1)}_{\text{max}}, \mathbf{Q}^{(1)}_{\text{max}})$ inscribed within this polytope can be obtained by solving the following problem:

$$\begin{aligned} \max \quad & \log\det \mathbf{Q}^{(1)}_{\text{max}} \\ \text{S.t.} \quad & \begin{pmatrix} (b_j - \mathbf{A}_j^T \mathbf{q}^{(1)}_{\text{max}})\mathbf{I}_n & \mathbf{Q}^{(1)}_{\text{max}}\mathbf{A}_j \\ \mathbf{A}_j^T \mathbf{Q}^{(1)}_{\text{max}} & (b_j - \mathbf{A}_j^T \mathbf{q}^{(\mathbf{1})}_{\text{max}}) \end{pmatrix} \succcurlyeq \mathbf{0}, \quad j = 1, 2, \ldots, m, \\ & \mathbf{Q}^{(1)}_{\text{max}} = \mathbf{Q}^{(1)T}_{\text{max}}, \qquad \mathbf{Q}^{(1)}_{\text{max}} \succ \mathbf{0}, \end{aligned} \tag{32}$$

where $\mathbf{A}_j^T$ is the j-th row of $\mathbf{A}$, and b_j is the j-th element of $\mathbf{b}$.

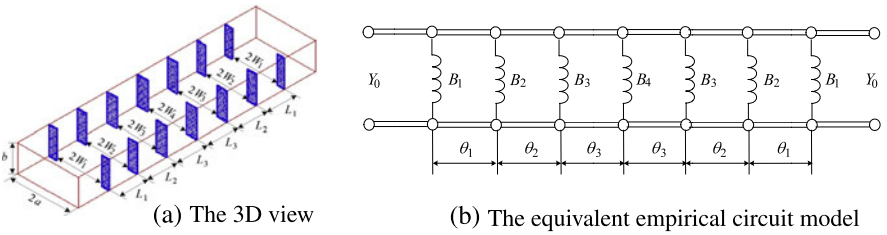
Proceeding with the last problem similarly to problem (27), both the ellipsoid center $\mathbf{q}^{(1)}_{\text{max}}$ and the ellipsoid matrix $\mathbf{B}^{(1)}_{\text{max}} = \mathbf{Q}^{(1)}_{\text{max}}\mathbf{Q}^{(1)T}_{\text{max}}$ can be obtained; hence the ellipsoid $\mathcal{E}^{(1)}_{\text{Max}}(\mathbf{q}^{(1)}_{\text{max}}, \mathbf{B}^{(1)}_{\text{max}})$ is obtained.

The previously stated two steps: polytope updating and maximum volume inscribed ellipsoid forming, are repeated until a suitable stopping criterion occurs. Hence, a polytope approximation for the feasible region, attached with the maximum volume ellipsoid $\mathcal{E}_{\text{MaxVE}}$ inscribed within this polytope, is obtained. Consequently, the maximum volume inscribed ellipsoid in the feasible region is obtained.

5.4.3 Design Centering of Six-Section H-Plane Waveguide Filter

The given method is applied for design centering of a six-section H-plane waveguide filter [39]. A waveguide with a width of 3.485 cm is used. The propagation mode is TE_{10} with a cutoff frequency of 4.3 GHz. The six-waveguide sections are separated by seven H-plane septa (as shown in Fig. 7) which have a finite thickness of 0.6223 mm. The design parameters $\mathbf{x}$ are the three waveguide section lengths L_1, L_2, and L_3 and the septa widths W_1, W_2, W_3, and W_4. The feasible region is constrained by the magnitude of the reflection coefficients $|S_{11}|$ at 44 frequency points $\{5.2, 5.3, \ldots, 9.5 \text{ GHz}\}$ as:

$$f_i\left(\mathbf{R}_f(\mathbf{x})\right) = \begin{cases} |S_{11}(\mathbf{x}, \omega_i)| \geq 0.85, & \omega_i \leq 5.2 \text{ GHz} \\ |S_{11}(\mathbf{x}, \omega_i)| \leq 0.16, & 5.4 \text{ GHz} \leq \omega_i \leq 9.0 \text{ GHz} \\ |S_{11}(\mathbf{x}, \omega_i)| \geq 0.5, & \omega_i \geq 9.5 \text{ GHz} \end{cases}$$



(a) The 3D view (b) The equivalent empirical circuit model

Fig. 7 The six-section H-plane waveguide filter

Table 5 Yield results for correlated parameter case

Covariance matrix	Initial yield	Final yield					
		Actual region		MinVE Approx.		MaxVE Approx.	
		q_{MinVE}	q_{MaxVE}	q_{MinVE}	q_{MaxVE}	q_{MinVE}	q_{MaxVE}
$\mathbf{B}_{\text{MinVE}}/49$	0 %	93 %	94 %	100 %	100 %	54 %	58 %
$\mathbf{B}_{\text{MaxVE}}/16$	0 %	100 %	100 %	100 %	100 %	92 %	96 %

Table 6 Yield results for independent parameter case

Parameter spreads	Initial yield	Final yield					
		Actual region		MinVE Approx.		MaxVE Approx.	
		q_{MinVE}	q_{MaxVE}	q_{MinVE}	q_{MaxVE}	q_{MinVE}	q_{MaxVE}
σ[a]	0 %	71 %	68 %	99 %	99 %	32 %	35 %
$\sigma/2$	0 %	99 %	97 %	100 %	100 %	87 %	93 %

[a] $\boldsymbol{\sigma} = 10^{-4}[0.762889\ 0.766507\ 0.797684\ 0.632623\ 0.584962\ 0.558823\ 0.551575]^T$

An empirical coarse model due to Marcuvitz [38], with lumped inductances and dispersive transmission line sections, is utilized. The simulation of the fine model is performed using High Frequency Structure Simulator (HFSS), starting with the point $\mathbf{x}_0 = 10^{-4}[160\ 160\ 165\ 135\ 120\ 115\ 115]^T$ mm with $\mathbf{B}_{\text{ini}} = 10^{-4} \times \text{I}_7$. Two SM iterations are needed to obtain the following two design centers:

$$\mathbf{q}_{\text{MinVE}} = 10^{-4}[159.703\ 161.876\ 165.426\ 134.022\ 122.086\ 116.969\ 115.662]^T,$$

$$\mathbf{q}_{\text{MaxVE}} = 10^{-4}[159.099\ 161.617\ 165.495\ 134.243\ 121.954\ 117.122\ 115.561]^T$$

The yield values are evaluated at $\mathbf{q}_{\text{init}}$, $\mathbf{q}_{\text{MinVE}}$ and $\mathbf{q}_{\text{MaxVE}}$ via the Monte Carlo method using 100 normally distributed samples for each yield evaluation. Tables 5 and 6 show the results for the correlated and independent cases, respectively.

Acknowledgements The authors would like to thank Prof. Slawomir Koziel, School of Science and Engineering, Reykjavik University, for his invitation to contribute to this book. The authors also would like to acknowledge the contributions to the original work by Prof. Hany Abdel-Malek,

Dr. Azza Rabie, Dr. Sameh Dakroury, Dr. Ahmed Abdel-Naby, and Eng. Ahmed El-Sharabasy, Faculty of Engineering, Cairo University, which have been reviewed in this chapter.

References

1. Abdel-Malek, H.L., Bandler, J.W.: Yield optimization for arbitrary statistical distributions: part I—theory. IEEE Trans. Circuits Syst. **27**, 245–253 (1980)
2. Abdel-Malek, H.L., Hassan, A.S.O.: The ellipsoidal technique for design centering and region approximation. IEEE Trans. Comput.-Aided Des. **10**, 1006–1014 (1991)
3. Abdel-Malek, H.L., Hassan, A.S.O., Bakr, M.H.: Statistical circuit design with the use of a modified ellipsoidal technique. Int. J. Microw. Millimeter Waves Comput.-Aided Eng. **7**, 117–129 (1997)
4. Abdel-Malek, H.L., Hassan, A.S.O., Bakr, M.H.: A boundary gradient search technique and its application in design centering. IEEE Trans. Comput.-Aided Des. **18**(11), 1654–1661 (1999)
5. Abdel-Malek, H.L., Hassan, A.S.O., Soliman, E.A., Dakroury, S.A.: The ellipsoidal technique for design centering of microwave circuits exploiting space-mapping interpolating surrogates. IEEE Trans. Microw. Theory Tech. **54**(10), 3731–3738 (2006)
6. Allen, P.E., Holberg, D.R.: CMOS Analog Circuit Design, 2nd edn. Oxford University Press, Oxford (2002)
7. Antreich, K.J., Graeb, H.E., Wieser, C.U.: Circuit analysis and optimization driven by worst-case distances. IEEE Trans. Comput.-Aided Des. **13**, 57–71 (1994)
8. Bakr, M.H., Bandler, J.W., Madsen, K., Søndergaard, J.: An introduction to the space mapping technique. Optim. Eng. **2**, 369–384 (2001)
9. Bandler, J.W., Abdel-Malek, H.L.: Optimal centering, tolerancing and yield determination via updated approximations and cuts. IEEE Trans. Circuits Syst. **25**, 853–871 (1978)
10. Bandler, J.W., Chen, S.H.: Circuit optimization: the state of the art. IEEE Trans. Microw. Theory Tech. **36**, 424–443 (1988)
11. Bandler, J.W., Seviora, R.E.: Computation of sensitivities for noncommensurate networks. IEEE Trans. Circuit Theory **CT-18**, 174–178 (1971)
12. Bandler, J.W., Zhang, Q.J., Song, J., Biernacki, R.M.: FAST gradient based yield optimization of nonlinear circuits. IEEE Trans. Microw. Theory Tech. **38**, 1701–1710 (1990)
13. Bandler, J.W., Biernacki, R.M., Chen, S.H., Grobelny, P.A., Hemmers, R.H.: Space mapping technique for electromagnetic optimization. IEEE Trans. Microw. Theory Tech. **42**, 2536–2544 (1994)
14. Bandler, J.W., Cheng, Q.S., Dakroury, S.A., Hailu, D.M., Madsen, K., Mohamed, A.S., Pedersen, F.: Space mapping interpolating surrogates for highly optimized EM-based design of microwave devices. In: IEEE MTT-S Int. Microwave Symp. Dig., Fort Worth, TX, vol. 3, pp. 1565–1568 (2004)
15. Bandler, J.W., Cheng, Q.S., Dakroury, S.A., Mohamed, A.S., Bakr, M.H., Madsen, K., Søndergaard, J.: Space mapping: the state of the art. IEEE Trans. Microw. Theory Tech. **52**, 337–361 (2004)
16. Broyden, C.G.: A class of methods for solving nonlinear simultaneous equations. Math. Comput. **19**, 577–593 (1965)
17. Cheng, Q.S., Bandler, J.W., Koziel, S., Bakr, M.H., Ogurtsov, S.: The state of the art of microwave CAD: EM-based optimization and modeling. Int. J. RF Microw. Comput.-Aided Eng. **20**, 475–491 (2010)
18. Conn, A.R., Toint, Ph.L.: An algorithm using quadratic interpolation for unconstrained derivative free optimization. In: Di Pillo, G., Giannes, F. (eds.) Nonlinear Optimization and Applications, pp. 27–47. Plenum Publishing, New York (1996)
19. Conn, A.R., Scheinberg, K., Toint, Ph.L.: A derivative free optimization algorithm in practice. In: Proceedings of 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO (1998)

20. De Boor, C., Ron, A.: Computational aspects of polynomial interpolation in several variables. Math. Comput. **58**, 705–727 (1992)
21. De Klerk, E.: Aspects of Semidefinite Programming: Interior-Point Algorithms and Selected Applications. Kluwer Academic, New York (2002)
22. Director, S.W., Hachtel, G.D., Vidigal, L.M.: Computationally efficient yield estimation procedures based on simplicial approximation. IEEE Trans. Circuits Syst. **25**, 121–130 (1978)
23. Elias, N.J.: Acceptance sampling: an efficient accurate method for estimating and optimizing parametric yield. IEEE J. Solid-State Circuits **29**, 323–327 (1994)
24. Fortran90 Software Repository. http://www.nag.co.uk/nagware/examples.asp
25. Giunta, A.A., Wojtkiewicz, S.F. Jr., Eldred, M.S.: Overview of modern design of experiments methods for computational simulations. In: Proceedings of the 41st AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV (2003). AIAA-2003-0649
26. Graeb, H.: Analog Design Centering and Sizing. Springer, Amsterdam (2007)
27. Hassan, A.S.O.: Normed distances and their applications in optimal circuit design. Optim. Eng. **4**, 197–213 (2003)
28. Hassan, A.S.O.: Design centering and region approximation via primal-dual interior-point technique. J. Eng. Appl. Sci. **51**(2), 195–212 (2004)
29. Hassan, A.S.O., Rabie, A.A.: Design centering using parallel-cuts ellipsoidal technique. J. Eng. Appl. Sci. **47**, 221–239 (2000)
30. Hassan, A.S.O., Abdel-Malek, H.L., Rabie, A.A.: Design centering and polyhedral region approximation via parallel-cuts ellipsoidal technique. Eng. Optim. **36**(1), 37–49 (2004)
31. Hassan, A.S.O., Abdel-Malek, H.L., Rabie, A.A.: None-derivative design centering algorithm using trust region optimization and variance reduction. Eng. Optim. **38**, 37–51 (2006)
32. Hassan, A.S.O., Mohamed, A.S., El-Sharabasy, A.Y.: Statistical microwave circuit optimization via a non-derivative trust region approach and space mapping surrogates. In: IEEE MTT-S Int. Microw. Symp. Dig., Baltimore, MD, USA (2011)
33. Hassan, A.S.O., Abdel-Naby, A.: A new hybrid method for optimal circuit design using semidefinite programming. Eng. Optim. 1–16 (2011)
34. Hocevar, D.E., Lightner, M.R., Trick, T.N.: A study of variance reduction techniques for estimating circuit yields. IEEE Trans. Comput.-Aided Des. **2**(3), 180–192 (1983)
35. Hocevar, D.E., Lightner, M.R., Trick, T.N.: An extrapolated yield approximation for use in yield maximization. IEEE Trans. Comput.-Aided Des. **3**, 279–287 (1984)
36. Keramat, M., Kielbasa, R.: A study of stratified sampling in variance reduction techniques for parametric yield estimations. IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process. **45**(5), 575–583 (1998)
37. Koziel, S., Bandler, J.W., Madsen, K.: A space-mapping frame work for engineering optimization: theory and implementation. IEEE Trans. Microw. Theory Tech. **54**(10), 3721–3730 (2006)
38. Marcuvitz, N.: Waveguide Handbook, 1st edn. McGraw-Hill, New York (1951)
39. Matthaei, G.L., Young, L., Jones, E.M.T.: Microwave Filters, Impedance-Matching Networks, and Coupling Structures, 1st edn. McGraw-Hill, New York (1964)
40. McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in analysis of output from a computer code. Technometrics **21**(2), 239–245 (1979)
41. Metropolis, N., Ulam, S.: The Monte-Carlo method. J. Am. Stat. Assoc. **44**, 335–341 (1949)
42. Moré, J.J., Sorensen, D.C.: Computing a trust region step. SIAM J. Sci. Stat. Comput. **4**(3), 553–572 (1983)
43. Powell, M.J.D.: UOBYQA. Unconstrained optimization by quadratic approximation. Math. Program. **92**, 555–582 (2002)
44. Powell, M.J.D.: The NEWUOA software for unconstrained optimization without derivatives. In: Di Pillo, G., Roma, M. (eds.) Large-Scale Nonlinear Optimization, pp. 255–297. Springer, New York (2006)
45. Powell, M.J.D.: Developments of NEWUOA for unconstrained minimization without derivatives. Technical Report, DAMTP 2007INA05, Department of Applied Mathematics and The-

oretical Physics, Cambridge University, UK (2007)
46. Sapatnekar, S.S., Vaidya, P.M., Kang, S.: Convexity-based algorithms for design centering. IEEE Trans. Comput.-Aided Des. **13**(12), 1536–1549 (1994)
47. Sauer, Th., Xu, Y.: On multivariate Lagrange interpolation. Math. Comput. **64**, 1147–1170 (1995)
48. Seifi, A., Ponnambalam, K., Vlach, J.: A unified approach to statistical design centering of integrated circuits with correlated parameters. IEEE Trans. Circuits Syst. **46**, 190–196 (1999)
49. Singhal, K., Pinel, J.F.: Statistical design centering and tolerancing using parametric sampling. IEEE Trans. Circuits Syst. **28**, 692–702 (1981)
50. Soliman, E.A., Bakr, M.H., Nikolova, N.K.: An adjoint variable method for sensitivity calculations of multiport devices. IEEE Trans. Microw. Theory Tech. **52**, 589–599 (2004)
51. Soliman, E.A.: Rapid frequency sweep technique for MoM planar solvers. IEE Proc. Microw. Antennas Propag. **151**, 277–282 (2004)
52. Soliman, E.A., Bakr, M.H., Nikolova, N.K.: Accelerated gradient-based optimization of planar circuits. IEEE Trans. Antennas Propag. **53**, 880–883 (2005)
53. Styblinski, M.A., Oplaski, L.J.: Algorithms and software tools for IC yield optimization based on fundamental fabrication parameters. IEEE Trans. Comput.-Aided Des. **5**, 79–89 (1986)
54. Toh, K.-C.: Primal-dual path-following algorithms for determinant maximization problems with linear matrix inequalities. Comput. Optim. Appl. **14**, 309–330 (1999)
55. Vandenberghe, L., Boyd, S., Wu, S.-P.: Determinant maximization with linear matrix inequality constraints. SIAM J. Matrix Anal. Appl. **19**, 499–533 (1998)
56. Vandenberghe, L., Boyd, S.: Applications of semidefinite programming. Appl. Numer. Math. **29**, 283–299 (1999)
57. Wojciechowski, J.M., Vlach, J.: Ellipsoidal method for design centering and yield estimation. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **12**, 1570–1579 (1993)
58. Wojciechowski, J., Opalski, L., Zantyniski, K.: Design centering using an approximation to the constraint region. IEEE Trans. Circuits Syst. **51**(3), 598–607 (2004)
59. Yu, T., Kang, S.M., Hajj, I.N., Trick, T.N.: Statistical performance modeling and parametric yield estimation of MOS VLSI. IEEE Trans. Comput.-Aided Des. **6**, 1013–1022 (1987)
60. Zaabab, A.H., Zhang, Q.J., Nakhla, M.: A neural network modeling approach to circuit optimization and statistical design. IEEE Trans. Microw. Theory Tech. **43**, 1349–1358 (1995)
61. Zhao, W., De, A., Donoro, D.G., Zhang, Y., Sarkar, T.K.: Antenna optimization by using NEWUOA. In: IEEE Antennas, Propag. Int. Symp., APSURSI 09 (2009)

Simulation-Driven Antenna Design Using Surrogate-Based Optimization

Slawomir Koziel, Stanislav Ogurtsov, and Leifur Leifsson

Abstract Accurate responses of antennas, in many cases, can be obtained only with discrete full-wave electromagnetic (EM) simulations. Therefore, contemporary antenna design strongly relies on these EM simulations. On the other hand, direct use of high-fidelity EM simulations in the design process, particularly for automated parameter optimization, often results in prohibitive computational costs. In this chapter, we illustrate how the designs of various antennas can be obtained efficiently using an automated surrogate-based optimization (SBO) methodology. The SBO techniques considered here include the adaptive design specification technique, variable-fidelity simulation-driven optimization, and shape-preserving response prediction. The essence of these techniques resides in shifting the optimization burden to a fast surrogate of the antenna structure, and using coarse-discretization EM models to configure the surrogate. A properly created and handled surrogate serves as a reliable prediction tool allowing satisfactory designs to be obtained at the cost of a few simulations of the high-fidelity antenna model. We also demonstrate the effect of the coarse-discretization model fidelity on the final design quality and the computational cost of the design process. Finally, we give an example of automatic management of the coarse model quality. Recommendations concerning the application of specific SBO techniques to antenna design are also presented.

Keywords Antenna design · Antenna optimization · Simulation-driven optimization · Electromagnetic (EM) simulation · Surrogate-based optimization (SBO) · Computer-aided design (CAD) · High-fidelity model · Coarse model · Coarse-discretization model

1 Introduction

Contemporary antenna design strongly relies on electromagnetic (EM) simulations [1]. For accurate evaluation of responses, antenna models should account

S. Koziel (✉) · S. Ogurtsov · L. Leifsson
Engineering Optimization & Modeling Center, School of Science and Engineering, Reykjavik University, Menntavegur 1, 101 Reykjavik, Iceland
e-mail: koziel@ru.is

S. Koziel, L. Leifsson (eds.), *Surrogate-Based Modeling and Optimization*,
DOI 10.1007/978-1-4614-7551-4_3,

for environmental features such as the installation fixture, connectors, and housing. Contemporary computational techniques—implemented in commercial simulation packages—are capable of obtaining quite accurate reflection and radiation antenna responses. However, full-wave simulations of realistic models are computationally expensive, and simulation even for a single combination of design parameters may take up to several hours. This computational cost poses a significant problem for antenna design.

A task of automated adjustment of antenna parameters can be formulated as an optimization problem with the objective function supplied by an EM solver [2]. However, most conventional optimization techniques—both gradient-based [3], e.g., conjugate-gradient, quasi-Newton, sequential quadratic programming, and derivative-free [4], e.g., Nelder–Mead and pattern search techniques—require large numbers of design simulations, each of which is already computationally expensive. As a consequence, the direct use of the EM solver to evaluate the high-fidelity antenna model in the optimization loop is often impractical due to the unacceptably high computational cost. Other obstacles for successful application of conventional optimization techniques to antenna design originate from the poor analytical properties of simulation-based objective functions (e.g., discontinuity, numerical noise). As a result, the practice of simulation-driven antenna design relies on repetitive parameter sweep. While this approach can be more reliable than brute-force antenna optimization, it is very laborious and time-consuming, and it does not guarantee optimal results. Also, only antenna designs with a limited number of parameters can be handled this way.

Adjoint sensitivities can substantially speed up microwave design optimization while using gradient-based algorithms [5] and [6]; however, adjoint sensitivities are not yet widespread in commercial EM solvers. Only CST Microwave Studio [7] and HFSS [8] have recently implemented this feature. Also, the use of adjoint sensitivities is limited by the numerical noise of the response.

Population-based techniques (metaheuristics) have recently become popular in solving certain antenna-design-related tasks [9, 10]. Methods such as genetic algorithms [11], particle swarm optimizers [12], or ant colony optimization [13] can alleviate certain problems (e.g., getting stuck in the local optimum). However, these methods are mainly applicable if objective function evaluation is very fast, for example, for synthesis of antenna array patterns [14]. The use of such techniques for simulation-based antenna design is questionable due to the large number of model evaluations required by metaheuristics.

In recent years, there has been a growing interest in surrogate-based optimization (SBO) methods [15–17], where direct optimization of the CPU-intensive full-wave EM model is replaced by iterative updating and reoptimization of a cheap and yet reasonably accurate representation of the antenna structure under consideration, called the surrogate model. There are many techniques exploiting both approximation surrogates, e.g., neural networks [18, 19], support vector regression [20, 21], radial basis functions [22], kriging [23, 24], fuzzy systems [25], and rational approximations [26], as well as physics-based surrogates (space mapping [15, 27–29], simulation-based tuning [30–32], manifold mapping [33], and shape-preserving response prediction [34]). Approximation models are fast and universal; however, they

are associated with a high initial cost (due to sampling of the design space and acquiring EM simulation data) and they are typically not suitable for ad hoc optimization. Techniques exploiting physics-based surrogates are particularly attractive because they are capable of yielding a satisfactory design using a very limited number of expensive high-fidelity EM simulations [15].

One of the most important assumptions to ensure efficiency of the SBO techniques exploiting physics-based surrogates is that the underlying low-fidelity model is computationally cheap. The most prominent technique of this kind is space mapping [34]. It originated in the area of microwave filter design, where this assumption is naturally satisfied with circuit equivalents [15] serving as low-fidelity models for filters. In the case of antennas, physics-based surrogates can be obtained from coarse-discretization EM simulations, as this is the only versatile way to create lower-fidelity antenna models. Unfortunately, these models may be relatively expensive. As a result, their evaluation cost cannot be neglected and may contribute considerably to the overall design expenses.

Therefore, the proper choice of the surrogate model fidelity (controlled, among other things, by the mesh density) is of great significance. On one hand, using a coarser low-fidelity model allows us to reduce its evaluation time; on the other hand, the coarser models are less accurate. As a result, a large number of iterations of the SBO algorithm may be necessary to yield a satisfactory design so that the total cost may be about the same or even higher than the total cost of an optimization algorithm employing only the finer model. Also, the SBO process may simply fail if the underlying low-fidelity model is not sufficiently accurate. For finer models, the individual evaluation time may be higher, but this is not directly translated into a higher total design cost because a smaller number of iterations may be sufficient to find a good design. In general, finding a good trade-off between the low-fidelity model speed and accuracy is not obvious.

In this chapter, we will review antenna design using physics-based surrogates originating from the coarse-mesh models. We also study the importance of a proper selection of the antenna model fidelity and its influence on performance of the surrogate-based design process in terms of the computational cost and design quality. Furthermore, we investigate the potential benefits of using several models of different fidelity in the same optimization run.

2 Surrogate-Based Design Optimization of Antennas

In this section, we consider a number of antenna design examples. In every example we describe the antenna structure under design, formulate the design problem, and outline the SBO technique that seems to be the most suitable to handle that particular antenna of interest. Results as well as design computational costs are provided.

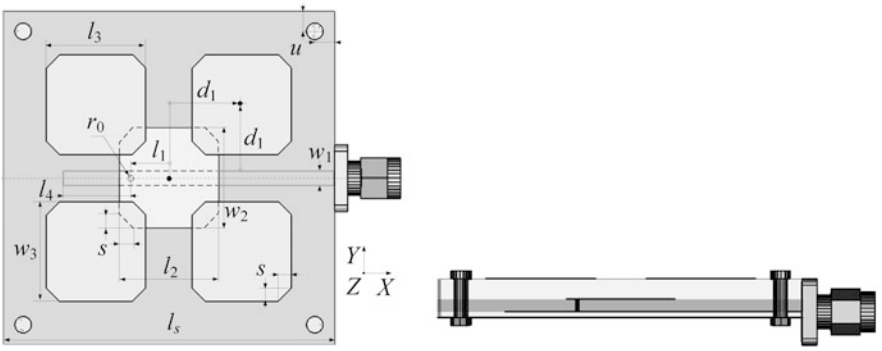


Fig. 1 Microstrip antenna [35]: top and side views, substrates shown transparent

2.1 Optimization of a Microstrip Composite Antenna Using the Multi-Fidelity Optimization Technique

Consider the composite microstrip antenna [35] shown in Fig. 1. The design variables are $\boldsymbol{x} = [l_1\ l_2\ l_3\ l_4\ w_2\ w_3\ d_1\ s]^T$. The multilayer substrate is $l_s \times l_s$ ($l_s =$ 30 mm). The antenna stack comprises a metal ground, RO4003 dielectric, signal trace, RO3006 dielectric with a through via connecting the trace to the driven patch, the driven patch, RO4003 dielectric, and four extra patches. The signal trace is terminated with an open-end stub. Feeding is with a 50 ohm SMA connector. The stack is fixed with four through bolts at the corners.

The final design is required to satisfy $|S_{11}| \leq -10$ dB for 3.1–4.8 GHz. The IEEE gain is required to be not less than 5 dB for the zero zenith angle over the whole frequency band of interest.

In this example, the antenna under design is of relatively complex composition; therefore, the choice of the mesh density for the coarse discretization model as well as other settings of the EM solver, here the CST MWS transient solver, strongly affect the total design optimization time. On the other hand, the computational cost of the model and its accuracy can be easily controlled by changing the discretization density. This feature has been exploited in the multi-fidelity optimization algorithm introduced in [36].

The multi-fidelity optimization is based on a family of coarse-discretization models $\{\boldsymbol{R}_{c.j}\}$, $j = 1, \ldots, K$, all evaluated by the same EM solver. Discretization of the model $\boldsymbol{R}_{c.j+1}$ is finer than that of the model $\boldsymbol{R}_{c.j}$, which results in better accuracy but also a longer evaluation time. In practice, the number of coarse-discretization models, K, is two or three.

When we have the optimized design $\boldsymbol{x}^{(K)}$ of the finest coarse-discretization model $\boldsymbol{R}_{c.K}$, the model is evaluated at all perturbed designs around $\boldsymbol{x}^{(K)}$, i.e., at $\boldsymbol{x}_k^{(K)} = [x_1^{(K)} \cdots x_k^{(K)} + \text{sign}(k) \cdot d_k \cdots x_n^{(K)}]^T$, $k = -n, -n+1, \ldots, n-1, n$. This data can be used to refine the final design without directly optimizing $\boldsymbol{R}_f$. Instead, an approximation model is set up and optimized in the neighborhood of $\boldsymbol{x}^{(K)}$ defined

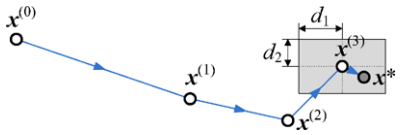


Fig. 2 Operation of the multi-fidelity design optimization procedure for three coarse-discretization models ($K = 3$). The design $\boldsymbol{x}^{(j)}$ is the optimal solution of the model $\boldsymbol{R}_{c.j}$, $j = 1, 2, 3$. A reduced second-order model $\boldsymbol{q}$ is set up in the neighborhood of $\boldsymbol{x}^{(K)}$ (*gray area*). The final design $\boldsymbol{x}^*$ is obtained by optimizing the model $\boldsymbol{q}$ as in (2)

as $[\boldsymbol{x}^{(K)} - \boldsymbol{d}, \boldsymbol{x}^{(K)} + \boldsymbol{d}]$, where $\boldsymbol{d} = [d_1\ d_2\ \ldots\ d_n]^T$, and $\boldsymbol{R}^{(k)}$ stands for $\boldsymbol{R}_{c.K}(\boldsymbol{x}_k^{(K)})$. The size of the neighborhood can be selected based on a sensitivity analysis of $\boldsymbol{R}_{c.1}$ (the cheapest of the coarse-discretization models); usually $\boldsymbol{d}$ equals 2 to 5 percent of $\boldsymbol{x}^{(K)}$. A reduced quadratic model $\boldsymbol{q}(\boldsymbol{x}) = [q_1\ q_2\ \ldots\ q_m]^T$ is used for approximation, where

$$\begin{aligned} q_j(\boldsymbol{x}) &= q_j\left([x_1 \ldots x_n]^T\right) \\ &= \lambda_{j.0} + \lambda_{j.1}x_1 + \cdots + \lambda_{j.n}x_n + \lambda_{j.n+1}x_1^2 + \cdots + \lambda_{j.2n}x_n^2. \end{aligned} \tag{1}$$

Coefficients $\lambda_{j.r}$, $j = 1, \ldots, m$, $r = 0, 1, \ldots, 2n$, can be uniquely obtained by solving the linear regression problem.

In order to account for unavoidable misalignment between $\boldsymbol{R}_{c.K}$ and $\boldsymbol{R}_f$, it is recommended to optimize a corrected model $\boldsymbol{q}(\boldsymbol{x}) + [\boldsymbol{R}_f(\boldsymbol{x}^{(K)}) - \boldsymbol{R}_{c.K}(\boldsymbol{x}^{(K)})]$ that ensures a zero-order consistency [37] between $\boldsymbol{R}_{c.K}$ and $\boldsymbol{R}_f$. The refined design can then be found as

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}^{(K)} - \boldsymbol{d} \le \boldsymbol{x} \le \boldsymbol{x}^{(K)} + \boldsymbol{d}} U\left(\boldsymbol{q}(\boldsymbol{x}) + \left[\boldsymbol{R}_f\left(\boldsymbol{x}^{(K)}\right) - \boldsymbol{R}_{c.K}\left(\boldsymbol{x}^{(K)}\right)\right]\right). \tag{2}$$

This kind of correction is also known as output space mapping [15]. If necessary, step (2) can be performed a few times starting from a refined design where each iteration requires only one evaluation of $\boldsymbol{R}_f$.

The multi-fidelity optimization procedure can be summarized as follows (the input arguments are initial design $\boldsymbol{x}^{(0)}$ and the number of coarse-discretization models K):

1. Set $j = 1$;
2. Optimize coarse-discretization model $\boldsymbol{R}_{c.j}$ to obtain a new design $\boldsymbol{x}^{(j)}$ using $\boldsymbol{x}^{(j-1)}$ as a starting point;
3. Set $j = j + 1$; if $j < K$ go to 2;
4. Obtain a refined design $\boldsymbol{x}^*$ as in (2);
5. END

Note that the original model $\boldsymbol{R}_f$ is only evaluated at the final stage (step 4). The operation of the algorithm in illustrated in Fig. 2. Coarse-discretization models can be optimized using any available algorithm.

Application of the multi-fidelity optimization methodology to this example can be outlined as follows. The initial design is set to $\boldsymbol{x}^{(0)} = [15\ 15\ 15\ 20\ -4\ 2\ 2]^T$ mm.

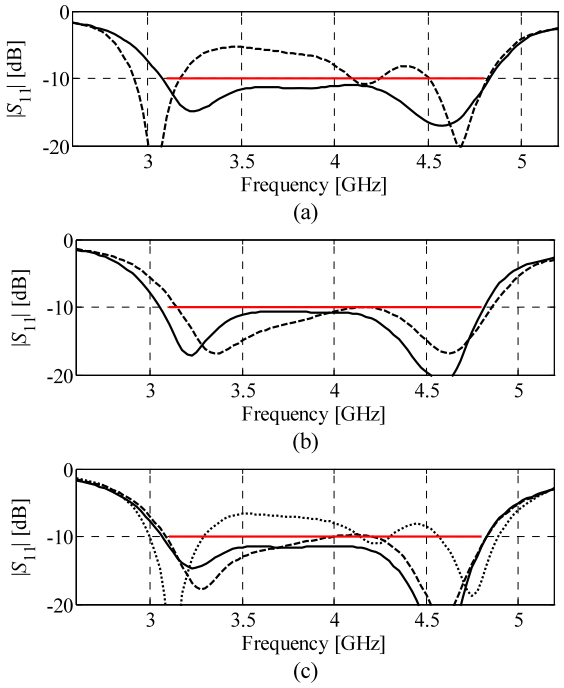


Fig. 3 Microstrip antenna: (**a**) model $\boldsymbol{R}_{c.1}$ at the initial design $\boldsymbol{x}^{(0)}$ (- - -) and at the optimized design $\boldsymbol{x}^{(1)}$ (—); (**b**) model $\boldsymbol{R}_{c.2}$ at $\boldsymbol{x}^{(1)}$ (- - -) and at its optimized design $\boldsymbol{x}^{(2)}$ (—); (**c**) model $\boldsymbol{R}_f$ at $\boldsymbol{x}^{(0)}$ ($\cdots$), at $\boldsymbol{x}^{(2)}$ (- - -), and at the refined final design $\boldsymbol{x}^*$ (—) [38]

Two coarse-discretization models are used: $\boldsymbol{R}_{c.1}$ (122,713 mesh cells at $\boldsymbol{x}^{(0)}$) and $\boldsymbol{R}_{c.2}$ (777,888 mesh cells). The evaluation times for $\boldsymbol{R}_{c.1}$, $\boldsymbol{R}_{c.2}$, and $\boldsymbol{R}_f$ (2,334,312 mesh cells) are 3 min, 18 min, and 160 min at $\boldsymbol{x}^{(0)}$, respectively. $|S_{11}|$ is the objective function with the goal of $|S_{11}| \leq -10$ dB for 3.1–4.8 GHz. An IEEE gain of not less than 5 dB for the zero elevation angle over the band is implemented as an optimization constraint.

Figure 3(a) shows the responses of $\boldsymbol{R}_{c.1}$ at $\boldsymbol{x}^{(0)}$ and at its optimal design $\boldsymbol{x}^{(1)}$. Figure 3(b) shows the responses of $\boldsymbol{R}_{c.2}$ at $\boldsymbol{x}^{(1)}$ and at its optimized design $\boldsymbol{x}^{(2)}$. Figure 3(c) shows the responses of $\boldsymbol{R}_f$ at $\boldsymbol{x}^{(0)}$, at $\boldsymbol{x}^{(2)}$, and at the refined design $\boldsymbol{x}^* = [14.87\ 13.95\ 15.4\ 13.13\ 20.87\ -5.90\ 2.88\ 0.68]^T$ mm ($|S_{11}| \leq -11.5$ dB for 3.1 GHz to 4.8 GHz) obtained in two iterations of the refinement step (2).

The design cost, shown in Table 1, corresponds to about 12 runs of the high-fidelity model $\boldsymbol{R}_f$. The antenna gain at the final design is shown in Fig. 4.

2.2 *Optimization of a Broadband Dielectric Resonator Antenna Using the Adaptively Adjusted Design Specifications Technique*

Consider the rotationally symmetric dielectric resonator antenna (DRA) [40] shown in Fig. 5. It comprises two annular ring dielectric resonators (DRs) with a relative

Table 1 Microstrip antenna: optimization cost [38]

Design step	Model evaluations	Computational cost	
		Absolute [hours]	Relative to $\boldsymbol{R}_f$
Optimization of $\boldsymbol{R}_{c.1}$	$125 \times \boldsymbol{R}_{c.1}$	6.3	2.6
Optimization of $\boldsymbol{R}_{c.2}$	$48 \times \boldsymbol{R}_{c.2}$	14.4	5.4
Setup of model $\boldsymbol{q}$	$17 \times \boldsymbol{R}_{c.2}$	5.1	1.9
Evaluation of $\boldsymbol{R}_f$	$2 \times \boldsymbol{R}_f$	5.3	2.0
Total design time	N/A	31.1	**11.9**

[a]Excludes $\boldsymbol{R}_f$ evaluation at the initial design

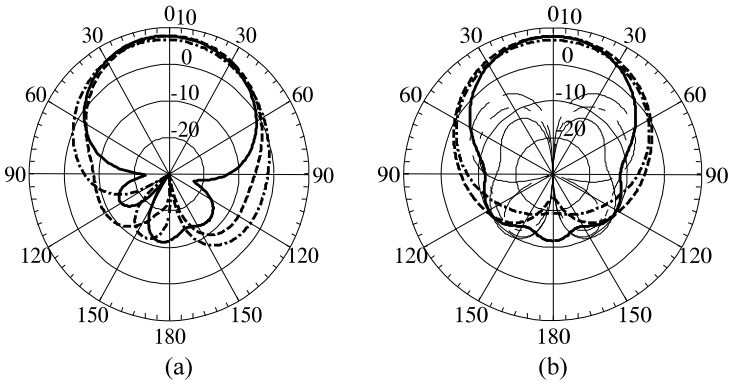


Fig. 4 Microstrip antenna gain [dBi] of the final design at 3.5 GHz (· — ·), 4.0 GHz (– –), and 4.5 GHz (—): (**a**) co-pol. in the E-plane (XOZ), with connector at 90° on the right; (**b**) x-pol., primary (*thick lines*) and co-pol. (*thin lines*) in the H-plane [39]

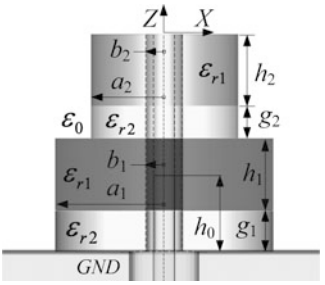


Fig. 5 Annular ring dielectric resonator antenna [40]: side view

permittivity, ε_{r1}, of 36; two supporting Teflon rings; a probe; and a cylindrical Teflon filling. The inner radius of the filling is the radius of the probe, 1.27 mm. The probe is an extension (h_0 above the ground) of the inner conductor of the input 50 ohm coaxial cable. The radius of each supporting ring equals that of the DR above it.

All metal parts are modeled as perfect electric conductors (PECs). The coax is also filled by Teflon. The ground is of infinite extent.

The design variables are the inner and outer radii of the DRs, heights of the DRs and the supporting rings, and the probe length, namely, $\boldsymbol{x} = [a_1\ a_2\ b_1\ b_2\ h_1\ h_2\ g_1\ g_2\ h_0]^T$. The design objective is $|S_{11}| \leq -20$ dB for 4 GHz to 6 GHz. A broadside gain of not less than 5 dBi is an optimization constraint.

Here, the overall shape of the low- and high-fidelity model responses is quite similar; therefore we use the adaptively adjusted design specifications (AADS) technique [41] that allows us to account for the misalignment between the models without actually adjusting the low-fidelity one. AADS consists of the following two steps that can be iterated if necessary:

1. Modify the original design specifications to account for the difference between the responses of the high-fidelity model $\boldsymbol{R}_f$ and the coarse-discretization model $\boldsymbol{R}_{\rm cd}$ at their characteristic points.
2. Obtain a new design by optimizing the low-fidelity model $\boldsymbol{R}_{\rm cd}$ with respect to the modified specifications.

As $\boldsymbol{R}_{\rm cd}$ is much faster than $\boldsymbol{R}_f$, the design process can be performed at low cost compared to direct optimization of $\boldsymbol{R}_f$. Figure 6 explains the idea of AADS using a bandstop filter example [41]. The design specifications are adjusted using characteristic points that should correspond to the design specification levels. They should also include local maxima/minima of the responses at which the specifications may not be satisfied.

It should be emphasized again that for the AADS technique there is no surrogate model configured from $\boldsymbol{R}_{\rm cd}$—the discrepancy between $\boldsymbol{R}_{\rm cd}$ and $\boldsymbol{R}_f$ is "absorbed" by the modified design specifications.

Figure 6(b) shows the characteristic points of $\boldsymbol{R}_f$ and $\boldsymbol{R}_{\rm cd}$. The design specifications are modified (mapped) so that the level of satisfying/violating the modified specifications by the $\boldsymbol{R}_{\rm cd}$ response corresponds to the satisfaction/violation levels of the original specifications by the $\boldsymbol{R}_f$ response (Figs. 6(b) and (c)). $\boldsymbol{R}_{\rm cd}$ is subsequently optimized with respect to the modified specifications, and the new design obtained this way is treated as an approximated solution to the original design problem. Typically, a substantial design improvement is observed after the first iteration. Additional iterations can bring further improvement.

The initial design is $\boldsymbol{x}^{\rm init} = [6.9\ 6.9\ 1.05\ 1.05\ 6.2\ 6.2\ 2.0\ 2.0\ 6.80]^T$. The high- and low fidelity models are evaluated using CST Microwave Studio ($\boldsymbol{R}_f$: 829,000 meshes at $\boldsymbol{x}^{\rm init}$, evaluation time 58 min, $\boldsymbol{R}_{\rm cd}$: 53,000 meshes at $\boldsymbol{x}^{\rm init}$, evaluation time 2 min).

The optimized design is found to be $\boldsymbol{x}^* = [5.9\ 5.9\ 1.05\ 1.55\ 7.075\ 7.2\ 4.5\ 1.0\ 8.05]^T$. It is obtained with three iterations of the AADS procedure. Significant improvement of the DRA's bandwidth is observed; the 48 % fractional bandwidth at -20 dB is shown in Fig. 7. The far-field response of the optimized DRA, shown in Fig. 8 at selected frequencies, stays at $\rm TM_{01\delta}$ DRA mode behavior over the 60 % bandwidth (on the -10 dB level). The total design cost is equivalent to about 11 evaluations of the high-fidelity DRA model. The design cost budget is listed in Table 2.

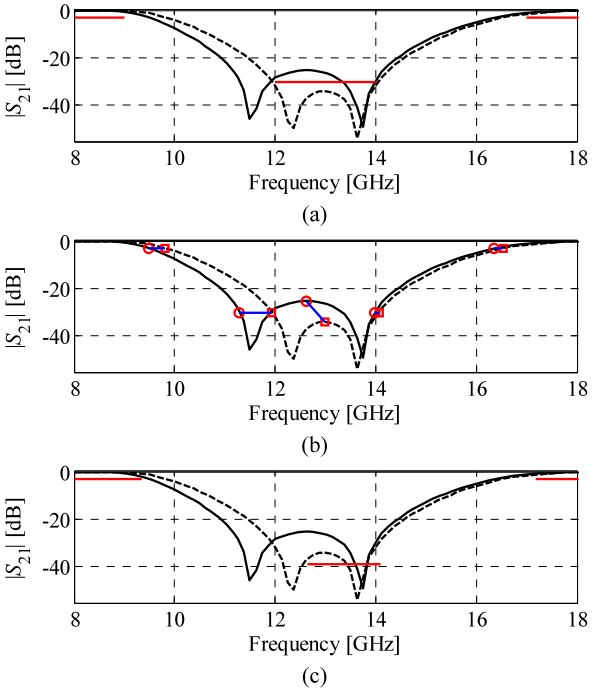


Fig. 6 AADS concept (responses of $\boldsymbol{R}_f$ (—) and $\boldsymbol{R}_{\text{cd}}$ (- - -)) [41]: (**a**) responses at the initial design and the original design specifications, (**b**) characteristic points of the responses corresponding to the specification levels (here, −3 dB and −30 dB) and to the local maxima, (**c**) responses at the initial design as well as the modified design specifications. The modification accounts for the discrepancy between the models so that optimizing $\boldsymbol{R}_{\text{cd}}$ with respect to the modified specifications approximately corresponds to optimizing $\boldsymbol{R}_f$ with respect to the original specifications

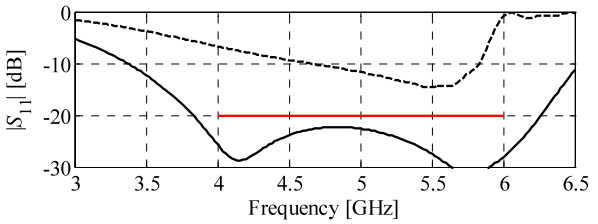


Fig. 7 DRA: fine model response at the initial (- - -) and the optimized design (—)

2.3 Design of UWB Antenna Using the Shape-Preserving Response Prediction Technique

Consider the planar antenna shown in Fig. 9. It consists of a planar dipole as the main radiator element and two additional strips. The design variables are $\boldsymbol{x} = [l_0\ w_0\ a_0\ l_p\ w_p\ s_0]^T$. Other dimensions are fixed as $a_1 = 0.5$ mm, $w_1 = 0.5$ mm,

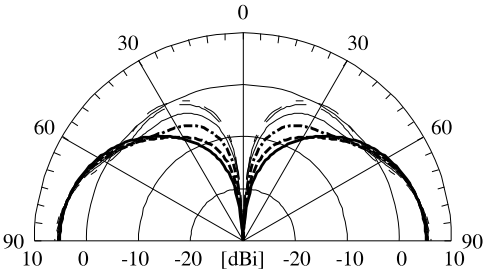


Fig. 8 DRA at the optimal design: gain in the elevation plane at 3.5 GHz (*thick* —), 4 GHz (*thick* – –), 4.5 GHz (*thick* · — ·), 5 GHz (*thin* –), 5.5 GHz (thin – –), and 6 GHz (*thin* · — ·)

Table 2 DRA: design optimization costs

Design step	Model evaluations	Computational cost	
		Absolute [hours]	Relative to $\boldsymbol{R}_f$
Optimization of $\boldsymbol{R}_{\text{cd}}$	$235 \times \boldsymbol{R}_{\text{cd}}$	7.8	8.1
Evaluation of $\boldsymbol{R}_f$	$3 \times \boldsymbol{R}_f$	2.9	3.0
Total design time	N/A	10.7	**11.1**

[a]Excludes $\boldsymbol{R}_f$ evaluation at the initial design

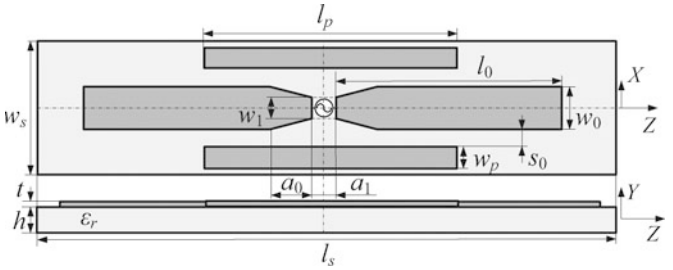


Fig. 9 UWB dipole antenna geometry: top and side views. The *dash-dotted lines* show the electric (YOZ) and the magnetic (XOY) symmetry walls. The 50 ohm source impedance is not shown

$l_s = 50$ mm, $w_s = 40$ mm, and $h = 1.58$ mm. The substrate material is Rogers RT5880.

The high-fidelity model $\boldsymbol{R}_f$ of the antenna structure (10,250,412 mesh cells at the initial design, evaluation time of 44 min) is simulated using the CST MWS transient solver. The design objective is to obtain $|S_{11}| \leq -12$ dB for 3.1 GHz to 10.6 GHz. The initial design is $\boldsymbol{x}^{\text{init}} = [20\ 10\ 1\ 10\ 82]^T$ mm. The low-fidelity model $\boldsymbol{R}_{\text{cd}}$ is also evaluated in CST but with coarser discretization (108,732 cells at $\boldsymbol{x}^{\text{init}}$, evaluated in 43 s).

For this example, the shapes of the low- and high-fidelity model response are similar (cf. Fig. 11(a)), which allows us to use the shape-preserving response

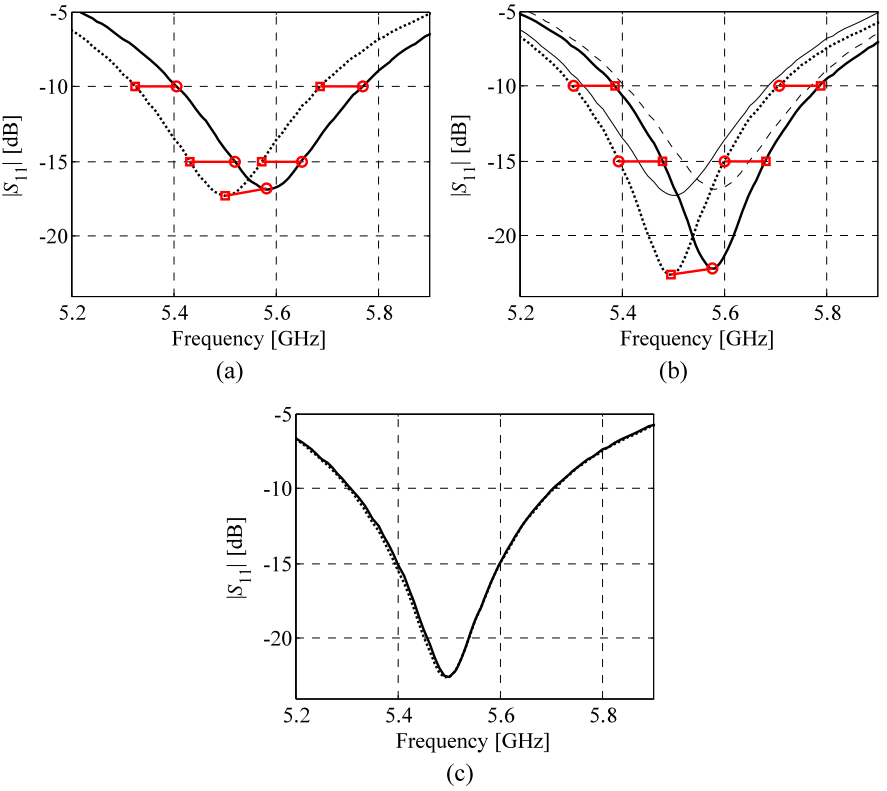


Fig. 10 SPRP concept. (**a**) Low-fidelity model response at the design $\boldsymbol{x}^{(i)}$, $\boldsymbol{R}_c(\boldsymbol{x}^{(i)})$ (*solid line*), the low-fidelity model response at $\boldsymbol{x}$, $\boldsymbol{R}_c(\boldsymbol{x})$ (*dotted line*), characteristic points of $\boldsymbol{R}_c(\boldsymbol{x}^{(i)})$ (*circles*) and $\boldsymbol{R}_c(\boldsymbol{x})$ (*squares*), and the translation vectors (*short lines*). (**b**) High-fidelity model response at $\boldsymbol{x}^{(i)}$, $\boldsymbol{R}_f(\boldsymbol{x}^{(i)})$ (*solid line*) and the predicted high-fidelity model response at $\boldsymbol{x}$ (*dotted line*) obtained using SPRP based on characteristic points of (**a**); characteristic points of $\boldsymbol{R}_f(\boldsymbol{x}^{(i)})$ (*circles*) and the translation vectors (*short lines*) were used to find the characteristic points (*squares*) of the predicted high-fidelity model response. (**c**) Low-fidelity model responses $\boldsymbol{R}_c(\boldsymbol{x}^{(i)})$ and $\boldsymbol{R}_c(\boldsymbol{x})$ are plotted using *thin solid* and *dotted line*, respectively

prediction (SPRP) technique [34] as the optimization engine. SPRP, unlike some other SBO techniques including space mapping, does not use any extractable parameters. As a result SPRP is typically very efficient: in many cases only two or three iterations are sufficient to yield a satisfactory design. SPRP assumes that the change of the high-fidelity model response due to the adjustment of the design variables can be predicted using the actual changes of the low-fidelity model response. Here, this property is ensured by the low-fidelity model being the coarse-mesh simulation of the same structure that represents the high-fidelity model.

The change of the low-fidelity model response can be described by the translation vectors corresponding to what are called the characteristic points of the model's re-

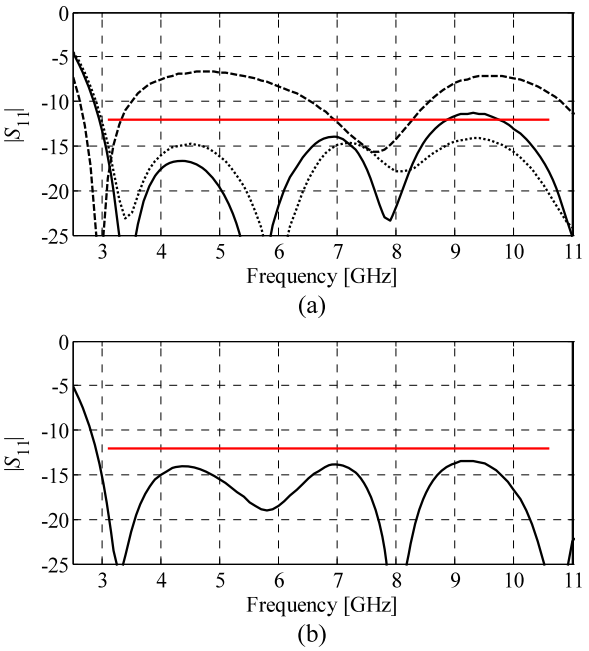


Fig. 11 UWB dipole antenna reflection response: (**a**) high-fidelity model response (*dashed line*) at the initial design $\boldsymbol{x}^{\text{init}}$, and high- (*solid line*) and low-fidelity (*dotted line*) model responses at the approximate low-fidelity model optimum $\boldsymbol{x}^{(0)}$; (**b**) high-fidelity model $|S_{11}|$ at the final design

sponse. These translation vectors are subsequently used to predict the change of the high-fidelity model response with the actual response of $\boldsymbol{R}_f$ at the current iteration point, $\boldsymbol{R}_f(\boldsymbol{x}^{(i)})$, treated as a reference.

Figure 10(a) shows an example low-fidelity model response, $|S_{11}|$ versus frequency, at the design $\boldsymbol{x}^{(i)}$, as well as the coarse model response at some other design $\boldsymbol{x}$. Circles denote characteristic points of $\boldsymbol{R}_c(\boldsymbol{x}^{(i)})$, selected here to represent $|S_{11}| = -10$ dB, $|S_{11}| = -15$ dB, and the local $|S_{11}|$ minimum. Squares denote corresponding characteristic points for $\boldsymbol{R}_c(\boldsymbol{x})$, while line segments represent the translation vectors ("shift") of the characteristic points of $\boldsymbol{R}_c$ when changing the design variables from $\boldsymbol{x}^{(i)}$ to $\boldsymbol{x}$.

The high-fidelity model response at $\boldsymbol{x}$ can be predicted using the same translation vectors applied to the corresponding characteristic points of the high-fidelity model response at $\boldsymbol{x}^{(i)}$, $\boldsymbol{R}_f(\boldsymbol{x}^{(i)})$. This is illustrated in Fig. 10(b). Figure 10(c) shows the predicted high-fidelity model response and the actual high-fidelity model response at $\boldsymbol{x}$. A rigorous and more detailed formulation of the SPRP technique can be found in [42].

For this example, the approximate optimum of $\boldsymbol{R}_{\text{cd}}$, $\boldsymbol{x}^{(0)} = [18.66\ 12.98\ 0.526\ 13.717\ 8.00\ 1.094]^T$ mm, is found as the first design step. The computational cost is 127 evaluations of $\boldsymbol{R}_{\text{cd}}$, which corresponds to about two evaluations of $\boldsymbol{R}_f$. Fig-

Table 3 UWB dipole antenna: optimization cost

Algorithm component	Number of model evaluations	Evaluation time	
		Absolute [min]	Relative to $\boldsymbol{R}_f$
Evaluation of $\boldsymbol{R}_{\text{cd}}$[a]	$233 \times \boldsymbol{R}_{\text{cd}}$	167	3.8
Evaluation of $\boldsymbol{R}_f$[b]	$3 \times \boldsymbol{R}_f$	132	3.0
Total optimization time	N/A	299	**6.8**

[a]Includes initial optimization of $\boldsymbol{R}_{\text{cd}}$ and optimization of SPRP surrogate

[b]Excludes evaluation of $\boldsymbol{R}_f$ at the initial design

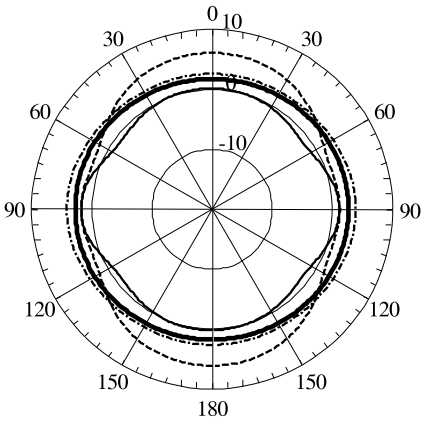


Fig. 12 UWB dipole at the final design: IEEE gain pattern (×-pol.) in the XOY plane at 4 GHz (*thick solid line*), 6 GHz (*dash-dotted line*), 8 GHz (*dashed line*), and 10 GHz (*solid line*)

ure 11(a) shows the reflection responses of $\boldsymbol{R}_f$ at both $\boldsymbol{x}^{\text{init}}$ and $\boldsymbol{x}^{(0)}$, as well as the response of $\boldsymbol{R}_{\text{cd}}$ at $\boldsymbol{x}^{(0)}$.

The final design $\boldsymbol{x}^{(2)} = [19.06\ 12.98\ 0.426\ 13.52\ 6.80\ 1.094]^T$ mm ($|S_{11}| \leq -13.5$ dB for 3.1 GHz to 10.6 GHz) is obtained after two iterations of the SPRP-based optimization with the total cost corresponding to about seven evaluations of the high-fidelity model (see Table 3). Figure 11(b) shows the reflection response and Fig. 12 shows the gain response of the final design $\boldsymbol{x}^{(2)}$.

2.4 Design of a Planar Antenna Array Using a Combination of Analytical and Coarse-Discretization Electromagnetic Models

The design of two-dimensional antenna arrays requires full-wave simulations, each of which is time-consuming due to the complexity and size of the antenna array under design as well as the electromagnetic (EM) interaction within the structure. Typically many EM simulations are necessary in the design process of a realistic

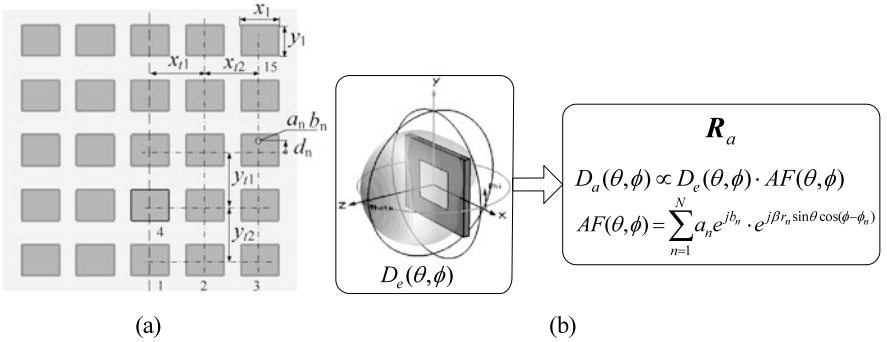


Fig. 13 Microstrip antenna array. (**a**) Front view of the EM models $\boldsymbol{R}_f$ and $\boldsymbol{R}_{\mathrm{cd}}$. The symmetry (magnetic) plane is shown with the vertical dashed line. (**b**) Analytical model of the planar array embedding the simulated radiation response of a single array element

array. Moreover, array design normally involves a large number of design variables, such as dimensions of the array elements, element spacing, location of feeds, excitation amplitudes and/or phases, and dimensions of the substrate and ground [1].

The array model based on the single element radiation response combined with the analytical array factor [43] cannot account for interelement coupling. In addition, this model produces inaccurate radiation responses in the directions off the main beam. Therefore, discrete EM simulations of the entire array are required; however, these simulations are computationally expensive when accurate. Consequently, using numerical optimization techniques to conduct the design process may be prohibitively expensive in terms of the CPU time. The use of coarse discretization for the whole array model can substantially relieve the computational load. However, the responses of such coarse-mesh models are typically noisy and often discontinuous, so that the optimization algorithm needs more objective function calls to find an improvement or it can even fail.

In order to reduce the computational cost of the array optimization process and make it robust, we apply surrogate-based optimization (SBO) [44] where we use an analytical model of the planar array embedding the simulated radiation response of a single array element, a coarse-discretization model of the entire array, and a fine model of the entire array. The design optimization example presented below describes and illustrates this approach.

Consider a planar microstrip array (Fig. 13) comprising 25 identical microstrip patches. The array is to operate at 10 GHz and have a linear polarization. Each patch is fed by a probe in the 50 ohm environment. The design tasks are as follows: to keep the lobe level below −20 dB for zenith angles off the main beam with a null-to-null width of 34°, i.e., off the sector of [−17°, 17°]; to maintain the peak directivity of the array at about 20 dBi; to have the direction of the maximum radiation perpendicular to the plane of the array; to have returning signals lower than −10 dB, all at 10 GHz. The initial dimensions of the elements, the microstrip patches, are 11 mm by 9 mm; a grounded layer of 1.58 mm thick RT/duroid 5880 is the substrate; the

lateral extension of the substrate/metal ground is set to a half of the patch size in a particular direction. The locations of the feeds at the initial design are at the center of the patch in the horizontal direction and 2.9 mm up off the center in the vertical direction, referring to Fig. 13(a). The symmetry of the array EM models is imposed as shown in Fig. 13(a).

The use of discrete EM models of the entire array is unavoidable here for several reasons, including the effect of element coupling on the reflection response and the requirement of minor lobe suppression. In the same time the evaluation time of the high-fidelity model of the array, $\boldsymbol{R}_f$, is around 20 minutes using the CST MWS transient solver, which makes its direct optimization impractical.

Even though we impose a symmetry on the array model and, therefore, restrict ourselves to adjusting distances between array components (x_{t1}, x_{t2}, y_{t1}, y_{t2}), patch dimensions (x_1, y_1), and the amplitudes ($a_1, \ldots a_{15}$) and/or phases ($b_1, \ldots b_{15}$) of the incident excitation signals, the number of design variables is still large for simulation-based design optimization. Therefore, we consider two design optimization cases: a design with nonuniform amplitude (and uniform phase) excitation where the design variables are $\boldsymbol{x} = [x_{t1}\ x_{t2}\ y_{t1}\ y_{t2}\ x_1\ y_1\ a_1\ \ldots\ a_{15}]^T$ and a design with nonuniform phase (uniform amplitude) excitation with $\boldsymbol{x} = [x_{t1}\ x_{t2}\ y_{t1}\ y_{t2}\ x_1\ y_1\ b_1\ \ldots\ b_{15}]^T$.

To evaluate the response of the array under design we adopt the following three EM models for it: a high-fidelity discrete EM model of the entire array, $\boldsymbol{R}_f$; a coarse-discretization EM model of the entire array $\boldsymbol{R}_{\rm cd}$ which is essentially a coarse-mesh version of $\boldsymbol{R}_f$ (evaluation time of $\boldsymbol{R}_{\rm cd}$ is about 1 min); and an analytical model of the array radiation pattern, $\boldsymbol{R}_a$ outlined in Fig. 13(b), which embeds the simulated radiation response of the single microstrip patch antenna. The use of these models in a developed SBO procedure is described in the following section.

Due to the high computational cost of evaluating the array, the design process exploits the SBO approach [45], where direct optimization of the array pattern is replaced by iterative correction and adjustment of the auxiliary models $\boldsymbol{R}_a$ and $\boldsymbol{R}_{\rm cd}$, described in the previous section.

The design procedure consists of the following two major stages:

Stage 1 (*pattern optimization*): In this stage, the design variables $\boldsymbol{x}$ are optimized in order to reduce the side low level according to the specifications. Starting from the initial design $\boldsymbol{x}^{(0)}$, the first approximation $\boldsymbol{x}^{(1)}$ of the optimum design is obtained by optimizing the analytical model $\boldsymbol{R}_a$. Further approximations $\boldsymbol{x}^{(i)}$, $i = 2, 3, \ldots$, are obtained as $\boldsymbol{x}^{(i)} = \text{argmin}\{\boldsymbol{x} : \boldsymbol{R}_a(\boldsymbol{x}) + [\boldsymbol{R}_f(\boldsymbol{x}^{(i-1)}) - \boldsymbol{R}_a(\boldsymbol{x}^{(i-1)})]\}$, i.e., by optimizing the analytical model $\boldsymbol{R}_a$ corrected using output space mapping [46] so that it matches the high-fidelity model exactly at the previous design $\boldsymbol{x}^{(i-1)}$. In practice, only two iterations are usually necessary to yield a satisfactory design. Note that each iteration of the above procedure requires only one evaluation of the high-fidelity model $\boldsymbol{R}_f$.

Stage 2 (*reflection adjustment*): In this stage, the coarse-discretization model $\boldsymbol{R}_{\rm cd}$ is used to correct the reflection of the array. Although we use the term "reflection response" and $|S_k|$ referring to returning signals at the feed points (ports), these signals include the effect of coupling due to simultaneous excitation of the elements.

In practice, after optimizing the pattern, the reflection responses are slightly shifted in frequency so that the minima of $|S_k|$ are not exactly at the required frequency (here, 10 GHz). The reflection responses can be shifted in frequency by adjusting the size of the patches, y_1 here. In order to find the appropriate change of y_1 we use the coarse-discretization model $\boldsymbol{R}_{\rm cd}$. Because both $\boldsymbol{R}_f$ and $\boldsymbol{R}_{\rm cd}$ are evaluated using the same EM solver, we assume that the frequency shift of reflection responses is similar for both models under the same change of the variable y_1, even though the responses themselves are not identical for $\boldsymbol{R}_f$ and $\boldsymbol{R}_{\rm cd}$ (in particular, they are shifted in frequency and the minimum levels of $|S_k|$ are typically different). By performing perturbation of y_1 using $\boldsymbol{R}_{\rm cd}$, one can estimate the change of y_1 in $\boldsymbol{R}_f$, necessary to obtain the required frequency shift of its reflection responses. This change would normally be very small so that it would not affect the array pattern in a substantial way. The computational cost of reflection adjustment using the method described here is only one evaluation of the high-fidelity model and one evaluation of the coarse-discretization model $\boldsymbol{R}_{\rm cd}$.

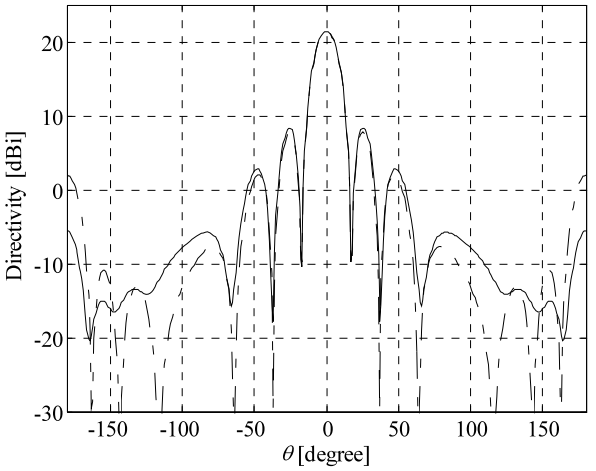
In the case of severe mismatch, the feed offsets d_n can also be used to adjust reflection; however, it was not necessary in the design cases considered in this work.

A starting point for the optimization procedure is chosen to be a uniform array, and the spacings x_{t1}, x_{t2}, y_{t1}, and y_{t2} are easily found using model $\boldsymbol{R}_a$ assuming $x_{t1} = x_{t2} = y_{t1} = y_{t2}$. The radiation response of the array at this design $\boldsymbol{x}^{(0)}$ is shown in Fig. 14. $\boldsymbol{x}^{(0)} = [x_{t1}\ x_{t2}\ y_{t1}\ y_{t2}\ x_1\ y_1\ a_1\ \ldots\ a_{15}\ b_1\ \ldots\ b_{15}]^T = [20\ 20\ 20\ 20\ 11\ 9\ 1\ \ldots\ 1\ 0\ \ldots 0]$ where the dimensional parameters are in millimeters, the excitation amplitudes are normalized, and the phase shifts are in degrees. The side lobe level of this design $\boldsymbol{x}^{(0)}$ is about -13 dB and the peak directivity of $\boldsymbol{x}^{(0)}$ is 21.4 dBi. The feed offset, d_n, shown in Fig. 13(a), is 2.9 mm for all patches.

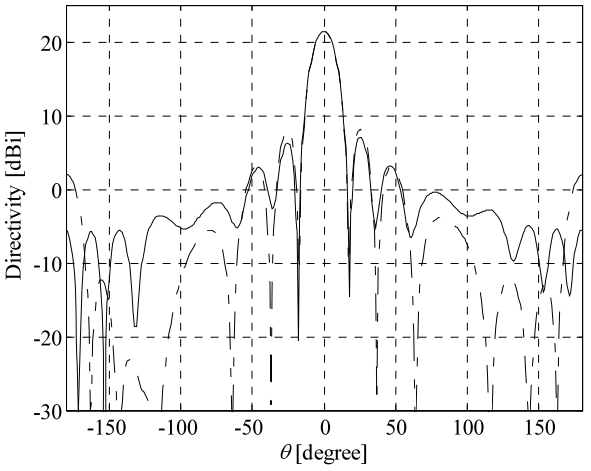
Design optimization with nonuniform amplitude excitation. Following the two-stage procedure described above, design optimization has been carried out with incident excitation amplitude as design variables. The cost of stage 1, directivity pattern optimization, is only three evaluations of $\boldsymbol{R}_f$ (the cost of optimizing the analytical $\boldsymbol{R}_a$ can be neglected). At stage 2, matching, we change the y-size of the patches, global parameter y_1 to 9.05 mm in order to move reflection responses to the left in frequency y_1. The cost of this step is $1 \times \boldsymbol{R}_{\rm cd} + 1 \times \boldsymbol{R}_f$.

The final design is found at $\boldsymbol{x}^* = [23.56\ 24.56\ 23.65\ 24.42\ 11.00\ 9.05\ 0.952\ 0.476\ 0.0982\ 0.982\ 0.946\ 0.525\ 1.000\ 0.973\ 0.932\ 0.994\ 0.936\ 0.529\ 0.858\ 0.594\ 0.0275]^T$. All excitation amplitudes are normalized to the maximum which is the amplitude of the seventh element located at the array center. The radiation response (directivity pattern cuts) and reflection response of the final design are shown in Fig. 15. The side lobe level of this design $\boldsymbol{x}^*$ is under -20 dB and the peak directivity of $\boldsymbol{x}^{(0)}$ is 21.8 dBi. The total cost of optimization is $1 \times \boldsymbol{R}_{\rm cd} + 4 \times \boldsymbol{R}_f$, that is, about $4 \times \boldsymbol{R}_f$.

Design optimization with nonuniform phase excitation. Another optimization case has been considered with the excitation phase shifts as design variables. The cost of stage 1, directivity pattern optimization, is again $3 \times \boldsymbol{R}_f$, and the cost of stage 2 is $1 \times \boldsymbol{R}_c + 1 \times \boldsymbol{R}_f$. The final design is found at $\boldsymbol{x}^* = [23.85\ 25.00\ 23.72\ 24.56\ 11.00\ 9.01\ 0.0\ -21.96\ 123.01\ 7.09\ -13.15\ 79.58\ 41.53$



(a)



(b)

Fig. 14 Microstrip antenna array of Fig. 13 at the initial (uniform) design $\boldsymbol{x}^{(0)}$, directivity pattern cuts at 10 GHz: (**a**) H-plane; (**b**) E-plane. EM model $\boldsymbol{R}_f$ (*solid lines*) and model $\boldsymbol{R}_a$ (*dash-dotted lines*)

37.33 0.51 24.75 65.94 −15.19 59.16 69.38 −67.62]^T where the phase shifts are in degrees and given relative to the first excitation element, which is shown in Fig. 13 and corresponds to the 0.0 entry in the vector $\boldsymbol{x}^*$. The radiation response (directivity pattern cuts) and reflection response of the final design are shown in Fig. 16. The side lobe level of this design $\boldsymbol{x}^*$ is about −19 dB and the peak directivity of $\boldsymbol{x}^{(0)}$ is 19.2 dBi. The total cost of optimization for this case is the same as in the previous example, i.e., around four evaluations of the high-fidelity model.

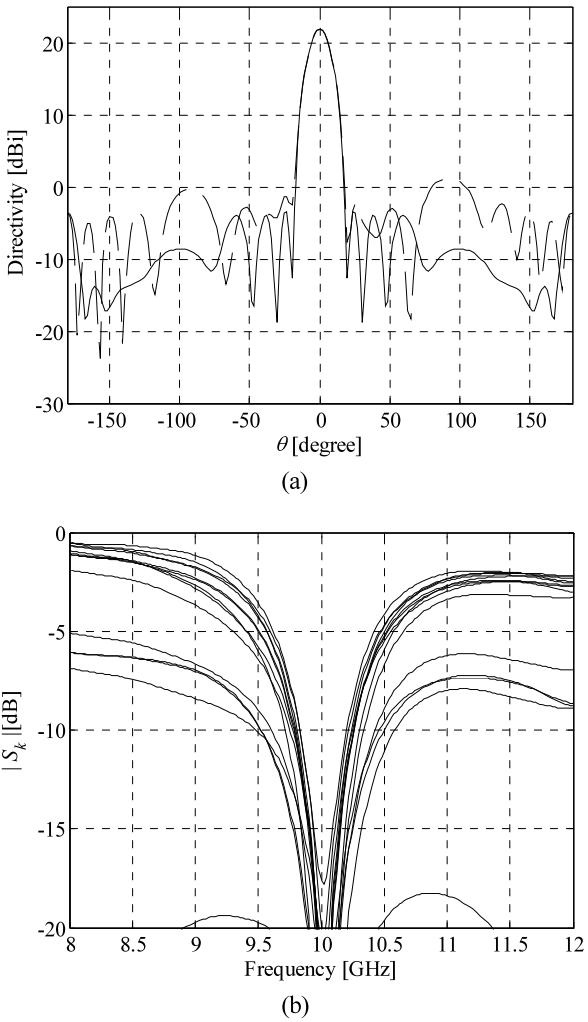


Fig. 15 Microstrip antenna array of Fig. 13 at the final design with nonuniform amplitude excitation: (**a**) directivity pattern cuts in the E- and H-planes at 10 GHz; (**b**) reflection responses of the array at the patch feeds

2.5 SBO Techniques for Antenna Design: Discussions and Recommendations

The SBO techniques presented in this section have proven to be computationally efficient for the design of different types of antennas. The typical computational cost of the design process expressed in terms of the number of equivalent high-fidelity model evaluations is comparable to the number of design variables, as demonstrated through examples. Here, we attempt to qualitatively compare these methods and

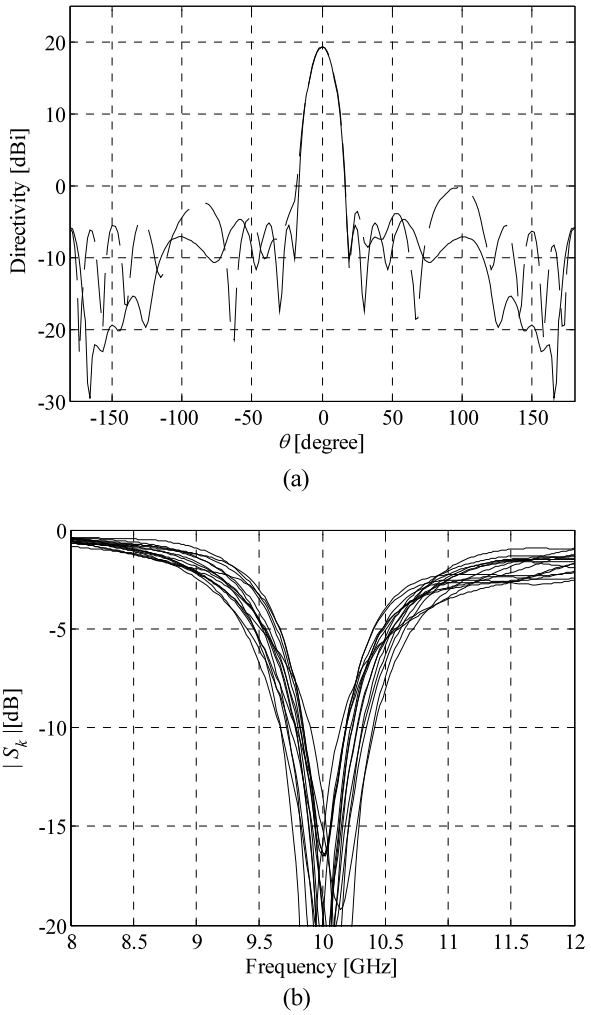


Fig. 16 Microstrip antenna array of Fig. 1 at the final design with nonuniform phase excitation: (**a**) directivity pattern cuts in the E- and H-planes at 10 GHz; (**b**) reflection responses of the array at the patch feeds

give some recommendations for the readers interested in using them in their research and design work.

The multi-fidelity approach applied in Sect. 2.1 is one of the most robust techniques, yet it is simple to implement. The only drawback is that it requires at least two low-fidelity models of different discretization density and some initial study of the model accuracy versus computational complexity. While the multi-fidelity technique will work with practically any setup, careful selection of the mesh density can reduce the computational cost of the optimization process considerably. More im-

plementation details and application examples of this technique can be found in [38] and [47].

Among the considered methods, the AADS technique of Sect. 2.2 is definitely the simplest for implementation, as it does not require any explicit correction of the low-fidelity model. Therefore, AADS can even be executed within any EM solver by modifying the design requirements and using its built-in optimization capabilities. On the other hand, AADS only works with minimax-like design specifications. Also, AADS requires the low-fidelity model to be relatively accurate so that the possible discrepancies between the low- and high-fidelity models can be accounted for by design specification adjustment. More implementation details and application examples of this technique to antenna design can be found in [48] and [49].

The SPRP technique of Sect. 2.3 does not use any extractable parameters. It assumes that the change of the high-fidelity model response due to the adjustment of the design variables can be predicted using the actual changes of the low-fidelity model response. SPRP is typically very efficient: in many cases only two or three iterations are sufficient to yield a satisfactory design [42].

Space mapping, discussed in Sect. 2.1 (at the last step of the variable-fidelity technique) and in Sect. 2.3, is a very generic method used to correct the low-fidelity model. In particular, it is able to work even if the low-fidelity model is rather inaccurate. On the other hand, space mapping requires some experience in selecting the proper type of surrogate model. More implementation details and application examples of this technique to antenna design can be found in [50] and [51] as well as in chapter Space Mapping for Electromagnetic-Simulation-Driven Design Optimization of this book.

As already mentioned, the low-fidelity model accuracy may be critical for the performance of the SBO algorithms. Using finer, i.e., more expensive but also more accurate, models generally reduces the number of SBO iterations necessary to find a satisfactory design; however, each SBO iteration turns to be more time-consuming. For coarser models, the cost of an SBO iteration is lower but the number of iterations may be larger, and for models that are too coarse, the SBO process may simply fail. The proper selection of the low-fidelity model "coarseness" may not be obvious beforehand. In most cases, it is recommended to use finer models rather than coarser ones to ensure good algorithm performance, even at the cost of some extra computational overhead.

The problem discussed in the previous paragraph can be considered in the wider context of model management, thus it may be beneficial to change the low-fidelity model coarseness during the SBO algorithm run. Typically, one starts from the coarser model in order to find an approximate location of the optimum design and switches to the finer model to increase the accuracy of the local search process without compromising the computational efficiency, e.g., as with the multi-fidelity technique of Sect. 2.1. Proper management of the model fidelity may result in further reduction of the design cost. The next section addresses this problem.

3 Model Fidelity Management for Cost-Efficient Surrogate-Based Design Optimization of Antennas

A proper choice of the surrogate model fidelity is a key factor that influences both the performance of the design optimization process and its computational cost. Here, we focus on a problem of proper surrogate model management. More specifically, we present a numerical study that aims for a trade-off between the design cost and reliability of the SBO algorithms. Our considerations are illustrated using several antenna design cases. Furthermore, we demonstrate that the use of multiple models of different fidelity may be beneficial for reducing the design cost while maintaining the robustness of the optimization process. Recommendations regarding the selection of the surrogate model coarseness are also given.

3.1 Coarse-Discretization Electromagnetic Simulations as Low-Fidelity Antenna Models

The only universal way of creating physics-based low-fidelity antenna models is through coarse-discretization EM simulations. This is particularly the case for wideband and ultra-wideband (UWB) antennas [52], as well as dielectric resonator antennas (DRAs) [53], to name just a few. Here, we assume that the low-fidelity model $\boldsymbol{R}_c$ is evaluated with the same EM solver as the high-fidelity model. The low-fidelity model can be created by reducing the mesh density compared to the high-fidelity one, as illustrated in Fig. 17. Other options of the low-fidelity model may include:

- Using a smaller computational domain with the finite-volume methods;
- Using low-order basis functions, e.g., with the moment method;
- Applying simple absorbing boundaries;
- Applying discrete sources rather than full-wave ports;
- Modeling metals with a perfect electric conductor;
- Neglecting the metallization thickness of traces, strips, and patches;
- Ignoring dielectric losses and dispersion.

Because of the possible simplifications, the low-fidelity model $\boldsymbol{R}_c$ is (typically 10 to 50 times) faster than $\boldsymbol{R}_f$ but not as accurate. Therefore, it cannot substitute for the high-fidelity model in design optimization. Obviously, making the low-fidelity model mesh coarser (and, perhaps, introducing other simplifications) results in a loss of accuracy but also in a shorter computational time. Figure 18 shows the plots illustrating the high- and low-fidelity model responses at a specific design for the antenna structure in Fig. 17, as well as the relationship between the mesh coarseness and the simulation time.

In Fig. 18, one can observe that the two "finest" coarse-discretization models (with ∼400,000 and ∼740,000 cells) properly represent the high-fidelity model response (shown as a thick solid line). The model with ∼270,000 cells can be considered as borderline. The two remaining models can be considered as too coarse,

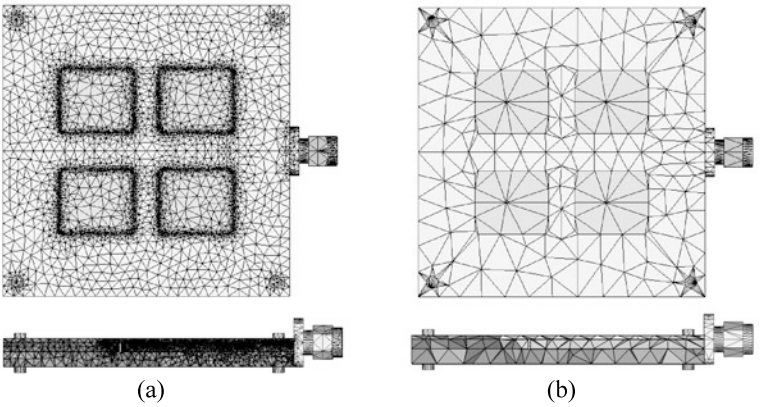


Fig. 17 A microstrip antenna [35]: (**a**) a high-fidelity EM model with a fine tetrahedral mesh, and (**b**) a low-fidelity EM model with a coarse tetrahedral mesh

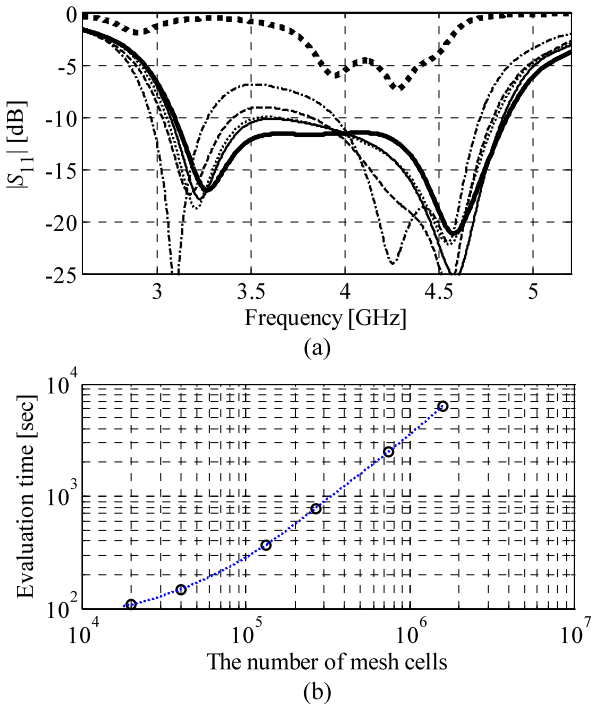
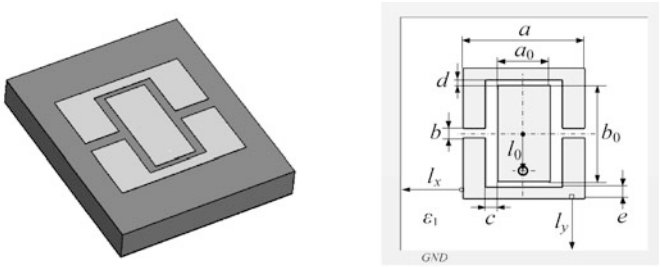


Fig. 18 An antenna of Fig. 17 evaluated with the CST MWS transient solver [7] at a selected design: (**a**) the reflection response with different discretization densities, 19,866 cells (■ ■ ■), 40,068 cells (· — ·), 266,396 cells (– –), 413,946 cells (· · ·), 740,740 cells (—), and 1,588,608 cells (—); and (**b**) the antenna evaluation time versus the number of mesh cells



(a) (b)

Fig. 19 Coax-fed microstrip antenna [54]: (**a**) 3D view, (**b**) top view

particularly the one with ∼20,000 cells; its response substantially deviates from that of the high-fidelity model.

We stress that, at the present stage of research, visual inspection of the model responses and the relationship between the high- and low-fidelity models is an important step in the model selection process. In particular, it is essential that the low-fidelity model capture all important features present in the high-fidelity one.

3.2 Selecting Model Fidelity: Design of Microstrip Antenna Using Frequency Scaling

We consider an antenna design case with the optimized designs found using an SBO algorithm of the following type. A generic SBO algorithm produced a series of approximate solutions to (1), $\boldsymbol{x}^{(i)}$, $i = 0, 1, \ldots$, as follows ($\boldsymbol{x}^{(0)}$ is the initial design) [15]:

$$\boldsymbol{x}^{(i+1)} = \arg\min_{\boldsymbol{x}} U\left(\boldsymbol{R}_s^{(i)}(\boldsymbol{x})\right), \tag{3}$$

where $\boldsymbol{R}_s^{(i)}$ is the surrogate model at iteration i. Typically, the surrogate model is updated after each iteration using the high-fidelity model data accumulated during the optimization process. Normally, the high-fidelity model is referred to rarely, in many cases only once per iteration, at a newly found design vector $\boldsymbol{x}^{(i+1)}$. This, in conjunction with the assumption that the surrogate model is fast, allows us to significantly reduce the computational cost of the design process when compared with direct solving of the original optimization problem.

Here we use three low-fidelity EM models of different mesh densities. We investigate the performance of the SBO algorithm working with these models in terms of the computational cost and the quality of the final design.

Consider the coax-fed microstrip antenna shown in Fig. 19 [54]. The design variables are $\boldsymbol{x} = [a\ b\ c\ d\ e\ l_0\ a_0\ b_0]^T$. The antenna is on 3.81 mm thick Rogers TMM4 substrate ($\varepsilon_1 = 4.5$ at 10 GHz); $l_x = l_y = 6.75$ mm. The ground plane is of infinite extent. The feed probe diameter is 0.8 mm. The connector's inner conductor is

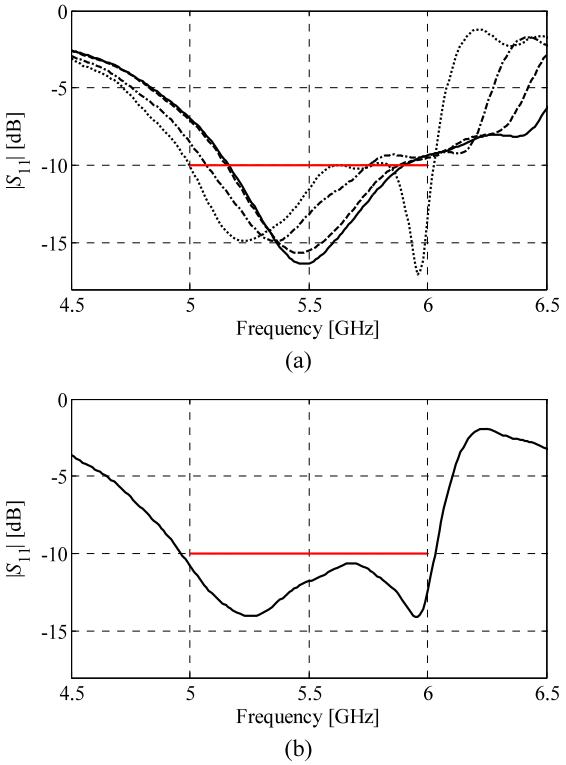


Fig. 20 Coax-fed microstrip antenna: (**a**) model responses at the initial design, $\boldsymbol{R}_{c1}$ $(\cdots)$, $\boldsymbol{R}_{c2}$ $(\cdot - \cdot -)$, $\boldsymbol{R}_{c3}$ (- - -), and $\boldsymbol{R}_f$ (—); (**b**) high-fidelity model response at the final design found using the low-fidelity model $\boldsymbol{R}_{c3}$

1.27 mm in diameter. The design specifications are $|S_{11}| \leq -10$ dB for 5 GHz to 6 GHz. The high-fidelity model $\boldsymbol{R}_f$ is evaluated with CST MWS transient solver [7] (704,165 mesh cells, evaluation time 60 min). We consider three coarse models: $\boldsymbol{R}_{c1}$ (41,496, 1 min), $\boldsymbol{R}_{c2}$ (96,096, 3 min), and $\boldsymbol{R}_{c3}$ (180,480, 6 min).

The initial design is $\boldsymbol{x}^{(0)} = [6\ 12\ 15\ 1\ 1\ 1\ 1\ -4]^T$ mm. Figure 20(a) shows the responses of all the models at the approximate optimum of $\boldsymbol{R}_{c1}$. The major misalignment between the responses is due to the frequency shift, so the surrogate is created here using frequency scaling as well as output space mapping [15] and [16]. The results, summarized in Table 4, indicate that the model $\boldsymbol{R}_{c1}$ is too inaccurate and the SBO design process using it fails to find a satisfactory design. The designs found with models $\boldsymbol{R}_{c2}$ and $\boldsymbol{R}_{c3}$ satisfy the specifications, and the cost of the SBO process using $\boldsymbol{R}_{c2}$ is slightly lower than that using $\boldsymbol{R}_{c3}$.

3.3 Coarse Model Management: Design of a Hybrid DRA

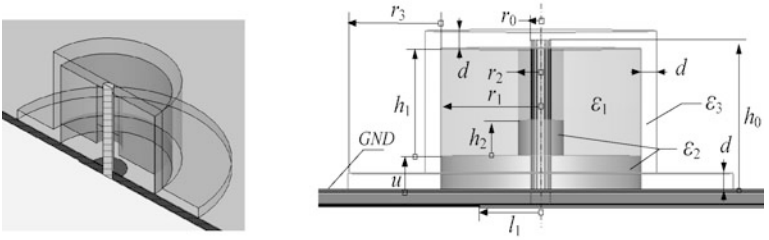
In this section, we again consider the use of low-fidelity models of various mesh densities for surrogate-based design optimization of the dielectric resonator antenna.

Table 4 Coax-fed microstrip antenna: design results

Low-fidelity model	Design cost: number of model evaluations[a]		Relative design cost[b]	Max $\|S_{11}\|$ for 5-to-6 GHz at final design
	$\boldsymbol{R}_c$	$\boldsymbol{R}_f$		
$\boldsymbol{R}_{c1}$	385	6	12.4	−8.0 dB
$\boldsymbol{R}_{c2}$	185	3	12.3	−10.0 dB
$\boldsymbol{R}_{c3}$	121	2	14.1	−10.7 dB

[a]Number of $\boldsymbol{R}_f$ evaluations is equal to the number of SBO iterations in (2)

[b]Equivalent number of $\boldsymbol{R}_f$ evaluations



(a) (b)

Fig. 21 Hybrid DRA: (**a**) 3D-cut view and (**b**) side view

We also investigate the potential benefits of using two models of different fidelity within a single optimization run.

Consider the hybrid DRA shown in Fig. 21. The DRA is fed by a 50 ohm microstrip terminated with an open-ended section. The microstrip substrate is 0.787 mm thick Rogers RT5880. The design variables are $\boldsymbol{x} = [h_0\ r_1\ h_1\ u\ l_1\ r_2]^T$. Other dimensions are fixed: $r_0 = 0.635$, $h_2 = 2$, $d = 1$, $r_3 = 6$, all in millimeters. The permittivity of the DRA core is 36, and the loss tangent is 10^{-4}, both at 10 GHz. The DRA support material is Teflon ($\varepsilon_2 = 2.1$), and the radome is of polycarbonate ($\varepsilon_3 = 2.7$ and $\tan\delta = 0.01$). The radius of the ground plane opening, shown in Fig. 21(b), is 2 mm.

The high-fidelity antenna model $\boldsymbol{R}_f(\boldsymbol{x})$ is evaluated using the time-domain solver of CST Microwave Studio [7] (∼1,400,000 meshes, evaluation time 60 min). The goal is to adjust the geometry parameters so that the following specifications are met: $|S_{11}| \leq -12$ dB for 5.15 GHz to 5.8 GHz. The initial design is $\boldsymbol{x}^{(0)} = [7.0\ 7.0\ 5.0\ 2.0\ 2.0\ 2.0]^T$ mm. We consider two auxiliary models of different fidelity, $\boldsymbol{R}_{c1}$ (∼45,000 meshes, evaluation time 1 min), and $\boldsymbol{R}_{c2}$ (∼300,000 meshes, evaluation time 3 min). We investigate the algorithm (2) using either one of these models or both ($\boldsymbol{R}_{c1}$ at the initial state and $\boldsymbol{R}_{c2}$ in the later stages). The surrogate model is constructed using both output space mapping and frequency scaling [15] and [16]. Figure 22(a) justifies the use of frequency scaling, which, due to the shape

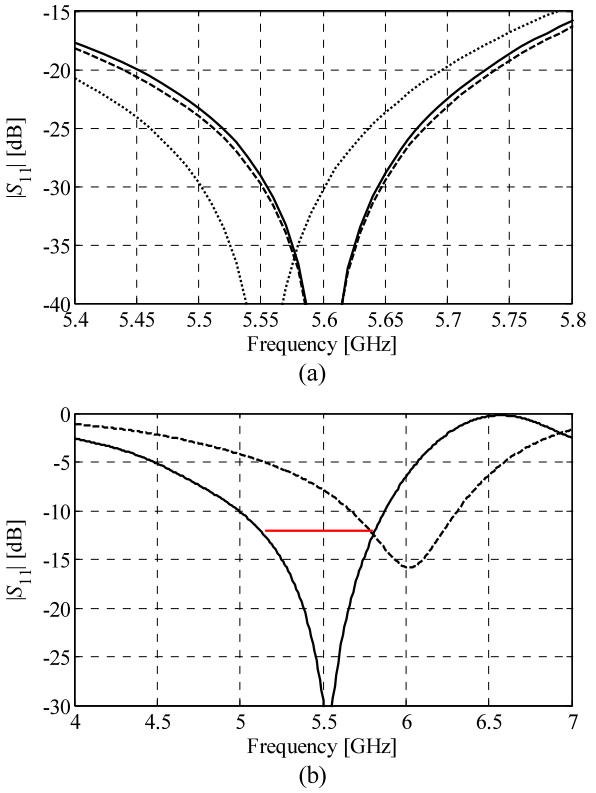


Fig. 22 Hybrid DRA: (**a**) high- (—) and low-fidelity model $\boldsymbol{R}_{c2}$ response at certain design before (· · · ·) and after (- - -) applying the frequency scaling, (**b**) high-fidelity model response at the initial design (- - -) and at the final design obtained using the SBO algorithm with the low-fidelity model $\boldsymbol{R}_{c2}$ (—)

similarity of the high- and low-fidelity model responses, allows substantial reduction of the misalignment between them.

The DRA design optimization has been performed three times: (i) with the surrogate constructed using $\boldsymbol{R}_{c1}$—cheaper but less accurate (case 1), (ii) with the surrogate constructed using $\boldsymbol{R}_{c2}$—more expensive but also more accurate (case 2), and (iii) with the surrogate constructed with $\boldsymbol{R}_{c1}$ at the first iteration and with $\boldsymbol{R}_{c2}$ for subsequent iterations (case 3). The last option allows us to more quickly locate the approximate high-fidelity model optimum and then refine it using the more accurate model. The number of surrogate model evaluations was limited to 100 (which involves the largest design change) in the first iteration and to 50 in the subsequent iterations (which require smaller design modifications).

Table 5 shows the optimization results for all three cases. Figure 22(b) shows the high-fidelity model response at the final design obtained using the SBO algorithm working with low-fidelity model $\boldsymbol{R}_{c2}$. The quality of the final designs found in all cases is the same. However, the SBO algorithm using the low-fidelity model $\boldsymbol{R}_{c1}$

Table 5 Hybrid DRA design results

Case	Number of iterations	Number of model evaluations[a]			Total design cost[b]	Max $\lvert S_{11}\rvert$ for 5.15 GHz to 5.8 GHz at final design
		$\boldsymbol{R}_{c1}$	$\boldsymbol{R}_{c2}$	$\boldsymbol{R}_f$		
1	4	250	0	4	8.2	−12.6 dB
2	2	0	150	2	9.5	−12.6 dB
3	2	100	50	2	6.2	−12.6 dB

[a]Number of $\boldsymbol{R}_f$ evaluations is equal to the number of SBO iterations

[b]Equivalent number of $\boldsymbol{R}_f$ evaluations

(case 1) requires more iterations than the algorithm using the model $\boldsymbol{R}_{c2}$ (case 3), because the latter is more accurate. In this particular case, the overall computational cost of the design process is still lower for $\boldsymbol{R}_{c1}$ than for $\boldsymbol{R}_{c2}$. On the other hand, the cheapest approach is case 2 when the model $\boldsymbol{R}_{c1}$ is utilized in the first iteration that requires the largest number of EM analyses, whereas the algorithm switches to $\boldsymbol{R}_{c2}$ in the second iteration, which allows us to both reduce the number of iterations and number of evaluations of $\boldsymbol{R}_{c2}$ at the same time. The total design cost is the lowest overall.

3.4 Discussion and Recommendations

The considerations and numerical results presented above allow us to draw some conclusions regarding the selection of model fidelity for surrogate-based antenna optimization. Using the cheaper (and less accurate) model may translate into a lower design cost; however, it also increases the risk of failure. Using the higher-fidelity model may increase the cost, but it definitely improves the robustness of the SBO design process and reduces the number of iterations necessary to find a satisfactory design. Visual inspection of the low- and high-fidelity model responses remains—so far—the most important way of accessing the model quality, which may also give a hint as to which type of model correction should be applied while creating the surrogate.

We can formulate the following rules of thumb and "heuristic" model selection procedure:

(i) An initial parametric study of low-fidelity model fidelity should be performed at the initial design in order to find the "coarsest" model that still adequately represents all the important features of the high-fidelity model response. The assessment should be done by visual inspection of the model responses, keeping in mind that the critical factor is not the absolute model discrepancy but the similarity of the response shape (e.g., even a relatively large frequency shift can be easily reduced by a proper frequency scaling).

(ii) When in doubt, it is safer to use a slightly finer low-fidelity model rather than a coarser one so that the potential cost reduction is not lost due to a possible algorithm failure to find a satisfactory design.
(iii) The type of misalignment between the low- and high-fidelity models should be observed in order to properly select the type of low-fidelity model correction while constructing the surrogate. The two methods considered here (additive response correction and frequency scaling) can be viewed as safe choices for most situations.

We emphasize that, for some antennas, such as some narrowband antennas or wideband traveling wave antennas, it is possible to obtain quite a good ratio between the simulation times of the high- and low-fidelity models (e.g., up to 50), because even for relatively coarse mesh, the low-fidelity model may still be a good representation of the high-fidelity one. For some structures (e.g., multi-resonant antennas), only much lower ratios (e.g., 5 to 10) may be possible, which would translate into lower design cost savings while using the SBO techniques.

4 Conclusion

Surrogate-based techniques for simulation-driven antenna design have been discussed, and it was demonstrated that optimized designs can be found at a low computational cost corresponding to a few high-fidelity EM simulations of the antenna structure. We also discussed an important trade-off between the computational complexity and accuracy of the low-fidelity EM antenna models and their effects on the performance of the surrogate-based optimization process. Recommendations regarding low-fidelity model selection were also formulated. We have demonstrated that by proper management of the models involved in the design process one can lower the overall optimization cost without compromising the final design quality. Further progress of the considered SBO techniques can be expected with their full automation, combination, and hybridization with adjoint sensitivities, as well as with metaheuristic algorithms.

References

1. Volakis, J.L. (ed.): Antenna Engineering Handbook, 4th edn. McGraw-Hill, New York (2007)
2. Special issue on synthesis and optimization techniques in electromagnetic and antenna system design. IEEE Trans. Antennas Propag. **55**, 518–785 (2007)
3. Wrigth, S.J., Nocedal, J.: Numerical Optimization. Springer, Berlin (1999)
4. Kolda, T.G., Lewis, R.M., Torczon, V.: Optimization by direct search: new perspectives on some classical and modern methods. SIAM Rev. **45**, 385–482 (2003)
5. Bandler, J.W., Seviora, R.E.: Wave sensitivities of networks. IEEE Trans. Microw. Theory Tech. **20**, 138–147 (1972)
6. Chung, Y.S., Cheon, C., Park, I.H., Hahn, S.Y.: Optimal design method for microwave device using time domain method and design sensitivity analysis—part II: FDTD case. IEEE Trans. Magn. **37**, 3255–3259 (2001)

7. CST Microwave Studio: CST AG, Bad Nauheimer Str. 19, D-64289, Darmstadt, Germany (2011)
8. HFSS: Release 13.0 ANSYS (2010). http://www.ansoft.com/products/hf/hfss/
9. Haupt, R.L.: Antenna design with a mixed integer genetic algorithm. IEEE Trans. Antennas Propag. **55**, 577–582 (2007)
10. Kerkhoff, A.J., Ling, H.: Design of a band-notched planar monopole antenna using genetic algorithm optimization. IEEE Trans. Antennas Propag. **55**, 604–610 (2007)
11. Pantoja, M.F., Meincke, P., Bretones, A.R.: A hybrid genetic algorithm space-mapping tool for the optimization of antennas. IEEE Trans. Antennas Propag. **55**, 777–781 (2007)
12. Jin, N., Rahmat-Samii, Y.: 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**, 3459–3468 (2005)
13. Halehdar, A., Thiel, D.V., Lewis, A., Randall, M.: Multiobjective optimization of small meander wire dipole antennas in a fixed area using ant colony system. Int. J. RF Microw. Comput.-Aided Eng. **19**, 592–597 (2009)
14. Jin, N., Rahmat-Samii, Y.: Analysis and particle swarm optimization of correlator antenna arrays for radio astronomy applications. IEEE Trans. Antennas Propag. **56**, 1269–1279 (2008)
15. Bandler, J.W., Cheng, Q.S., Dakroury, S.A., Mohamed, A.S., Bakr, M.H., Madsen, K., Søndergaard, J.: Space mapping: the state of the art. IEEE Trans. Microw. Theory Tech. **52**, 337–361 (2004)
16. Koziel, S., Bandler, J.W., Madsen, K.: A space mapping framework for engineering optimization: theory and implementation. IEEE Trans. Microw. Theory Tech. **54**, 3721–3730 (2006)
17. Koziel, S., Echeverría-Ciaurri, D., Leifsson, L.: Surrogate-based methods. In: Koziel, S., Yang, X.S. (eds.) Computational Optimization, Methods and Algorithms. Studies in Computational Intelligence, pp. 33–60. Springer, Berlin (2011)
18. Rayas-Sánchez, J.E.: EM-based optimization of microwave circuits using artificial neural networks: the state-of-the-art. IEEE Trans. Microw. Theory Tech. **52**, 420–435 (2004)
19. Kabir, H., Wang, Y., Yu, M., Zhang, Q.J.: Neural network inverse modeling and applications to microwave filter design. IEEE Trans. Microw. Theory Tech. **56**, 867–879 (2008)
20. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Stat. Comput. **14**, 199–222 (2004)
21. Meng, J., Xia, L.: Support-vector regression model for millimeter wave transition. Int. J. Infrared Millim. Waves **28**, 413–421 (2007)
22. Buhmann, M.D., Ablowitz, M.J.: Radial Basis Functions: Theory and Implementations. Cambridge University Press, Cambridge (2003)
23. Simpson, T.W., Peplinski, J., Koch, P.N., Allen, J.K.: Metamodels for computer-based engineering design: survey and recommendations. Eng. Comput. **17**, 129–150 (2001)
24. Forrester, A.I.J., Keane, A.J.: Recent advances in surrogate-based optimization. Prog. Aerosp. Sci. **45**, 50–79 (2009)
25. Miraftab, V., Mansour, R.R.: EM-based microwave circuit design using fuzzy logic techniques. IEE Proc., Microw. Antennas Propag. **153**, 495–501 (2006)
26. Shaker, G.S.A., Bakr, M.H., Sangary, N., Safavi-Naeini, S.: Accelerated antenna design methodology exploiting parameterized Cauchy models. Prog. Electromagn. Res. **18**, 279–309 (2009)
27. Amari, S., LeDrew, C., Menzel, W.: Space-mapping optimization of planar coupled-resonator microwave filters. IEEE Trans. Microw. Theory Tech. **54**, 2153–2159 (2006)
28. Quyang, J., Yang, F., Zhou, H., Nie, Z., Zhao, Z.: Conformal antenna optimization with space mapping. J. Electromagn. Waves Appl. **24**, 251–260 (2010)
29. Koziel, S., Cheng, Q.S., Bandler, J.W.: Space mapping. IEEE Microw. Mag. **9**, 105–122 (2008)
30. Swanson, D., Macchiarella, G.: Microwave filter design by synthesis and optimization. IEEE Microw. Mag. **8**, 55–69 (2007)
31. Rautio, J.C.: Perfectly calibrated internal ports in EM analysis of planar circuits. In: IEEE MTT-S Int. Microwave Symp. Dig., Atlanta, GA, pp. 1373–1376 (2008)

32. Cheng, Q.S., Rautio, J.C., Bandler, J.W., Koziel, S.: Progress in simulator-based tuning—the art of tuning space mapping. IEEE Microw. Mag. **11**, 96–110 (2010)
33. Echeverria, D., Hemker, P.W.: Space mapping and defect correction. Comput. Methods Appl. Math. **5**, 107–136 (2005)
34. Koziel, S.: Shape-preserving response prediction for microwave design optimization. IEEE Trans. Microw. Theory Tech. **58**, 2829–2837 (2010)
35. Chen, Z.N.: Wideband microstrip antennas with sandwich substrate. IET Microw. Antennas Propag. **2**, 538–546 (2008)
36. Koziel, S., Ogurtsov, S.: Robust multi-fidelity simulation-driven design optimization of microwave structures. In: IEEE MTT-S Int. Microwave Symp. Dig., pp. 201–204 (2010)
37. Alexandrov, N.M., Dennis, J.E., Lewis, R.M., Torczon, V.: A trust region framework for managing use of approximation models in optimization. Struct. Multidiscip. Optim. **15**, 16–23 (1998)
38. Koziel, S., Ogurtsov, S.: Antenna design through variable-fidelity simulation-driven optimization. In: Loughborough Antennas & Propagation Conference, LAPC 2011, IEEEXplore (2011)
39. Koziel, S., Ogurtsov, S.: Computationally efficient simulation-driven antenna design using coarse-discretization electromagnetic models. In: IEEE Int. Symp. Antennas Propag., pp. 2928–2931 (2011)
40. Shum, S., Luk, K.: Stacked anunular ring dielectric resonator antenna excited by axisymmetric coaxial probe. IEEE Trans. Microw. Theory Tech. **43**, 889–892 (1995)
41. Koziel, S.: Efficient optimization of microwave structures through design specifications adaptation. In: Proc. IEEE Antennas Propag. Soc. International Symposium (APSURSI), Toronto, Canada (2010)
42. Koziel, S.: Shape-preserving response prediction for microwave design optimization. IEEE Trans. Microw. Theory Tech. **58**, 2829–2837 (2010)
43. Balanis, C.A.: Antenna Theory, 3rd edn. Wiley-Interscience, New York (2005)
44. Koziel, S., Ogurtsov, S.: Simulation-driven design in microwave engineering: methods. In: Koziel, S., Yang, X.S. (eds.) Computational Optimization, Methods and Algorithms. Studies in Computational Intelligence. Springer, Berlin (2011)
45. Queipo, N.V., Haftka, R.T., Shyy, W., Goel, T., Vaidynathan, R., Tucker, P.K.: Surrogatebased analysis and optimization. Prog. Aerosp. Sci. **41**, 1–28 (2005)
46. Bandler, J.W., Koziel, S., Madsen, K.: Space mapping for engineering optimization. SIAG/Optimization Views-and-News Special Issue on Surrogate/Derivative-free Optimization **17**, 19–26 (2006)
47. Koziel, S., Ogurtsov, S., Leifsson, L.: Variable-fidelity simulation-driven design optimisation of microwave structures. Int. J. Math. Model. Numer. Optim. **3**, 64–81 (2012)
48. Ogurtsov, S., Koziel, S.: Optimization of UWB planar antennas using adaptive design specifications. In: Proc. the 5th European Conference on Antennas and Propagation, EuCAP, pp. 2216–2219 (2011)
49. Ogurtsov, S., Koziel, S.: Design optimization of a dielectric ring resonator antenna for matched operation in two installation scenarios. In: Proc. International Review of Progress in Applied Computational Electromagnetics, ACES, pp. 424–428 (2011)
50. Koziel, S., Ogurtsov, S.: Rapid design optimization of antennas using space mapping and response surface approximation models. Int. J. RF Microw. Comput.-Aided Eng. **21**, 611–621 (2011)
51. Ogurtsov, S., Koziel, S.: Simulation-driven design of dielectric resonator antenna with reduced board noise emission. In: IEEE MTT-S Int. Microwave Symp. Dig. (2011)
52. Schantz, H.: The Art and Science of Ultrawideband Antennas. Artech House, New York (2005)
53. Petosa, A.: Dielectric Resonator Antenna Handbook. Artech House, New York (2007)
54. Wi, S.-H., Lee, Y.-S., Yook, J.-G.: Wideband microstrip patch antenna with U-shaped parasitic elements. IEEE Trans. Antennas Propag. **55**, 1196–1199 (2007)

Practical Application of Space Mapping Techniques to the Synthesis of CSRR-Based Artificial Transmission Lines

Ana Rodríguez, Jordi Selga, Ferran Martín, and Vicente E. Boria

Abstract Artificial transmission lines based on metamaterial concepts have been attracting increasing interest from the scientific community. The synthesis process of this type of artificial line is typically a complex task, due to the number of design parameters involved and their mutual dependence. Space mapping techniques are revealed to be very useful for automating the synthesis procedure of these kinds of structures. In this chapter, a review of their application to the automated synthesis of microstrip lines loaded with complementary split ring resonators (CSRRs), either with or without series capacitive gaps, will be presented. The most critical points related to the implementation of these space mapping techniques are discussed in detail. Different examples to illustrate and prove the usefulness of this synthesis methodology are presented.

Keywords Aggressive space mapping · Computer aided design (CAD) · Planar microwave circuits · Complementary split resonators · Artificial transmission lines

1 Introduction

Space mapping (SM) techniques and SM-based surrogate (modeling) have been widely used in the design of many microwave components [1], but their application

A. Rodríguez (✉) · V.E. Boria
Departamento de Comunicaciones-iTEAM, Universitat Politècnica de València, 46022 Valencia, Spain
e-mail: amrodri@iteam.upv.es

V.E. Boria
e-mail: vboria@dcom.upv.es

J. Selga · F. Martín
CIMITEC, Departament d'Enginyeria Electrònica, Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain

F. Martín
e-mail: Ferran.Martin@uab.es

V.E. Boria
Aurora Software and Testing S.L., Valencia, Spain

S. Koziel, L. Leifsson (eds.), *Surrogate-Based Modeling and Optimization*,
DOI 10.1007/978-1-4614-7551-4_4,

to the synthesis of artificial transmission lines is still marginal. Metamaterial transmission lines can be a very interesting alternative to conventional lines for many practical microwave devices [2, 3], since more compact solutions (compatible with planar manufacturing processes) with higher degrees of design flexibility can be obtained. Furthermore, metamaterial transmission lines can also provide many other unique properties not achievable with conventional lines, which include negative or low values of permittivity, permeability, and index of refraction, controllable dispersion, and tailored characteristic impedance values [4]. Their commercial impact is still not very relevant, due the complexity of the related synthesis procedures used nowadays. These procedures are mainly based on the engineer's experience with the help of full-wave electromagnetic (EM) simulators and parameter extraction methods. Hence, the motivation of this chapter is to review some promising advances made recently in the field, which can simplify and speed up the process of artificial line synthesis, and that it is hoped will be of interest to the readers.

There are two main approaches for implementing metamaterial transmission lines. One strategy consists in loading a host transmission line with series capacitances and shunt inductances and, as a result, presenting a composite right/left-handed behavior. In order to have propagation, the series and shunt reactances of the equivalent circuit model have to present opposite signs. Depending on the frequency range, either the loading elements (capacity gaps, strips, vias) or the host line will determine the propagation characteristics. The mushroom structure, originally introduced by Sievenpiper et al. [5] and used often in low-profile antennas in order to improve the gain and directivity, is a good example of this strategy, which is usually known as the CL-loaded approach. The CL-loaded term means that the host transmission line is loaded with series capacitances and shunted inductances.

The alternative technique to obtain metamaterial transmission lines consists of etching electrically small (subwavelength) resonators to the host transmission line (TL): this is called the resonant-type approach [6]. All these artificial transmission lines are also identified as composite right/left-handed (CRLH) lines, since they exhibit left-handed (backward) wave propagation at low frequencies and right-handed (forward) wave propagation at high frequencies [7]. Depending on different aspects, such as the kind of resonator, how it is coupled to the line, and the additional elements it is combined with, it is possible to design many different transmission lines [6, 8]. The split ring resonator (SRR) is probably the most widely used common element, and it is basically composed of two concentric metallic rings with splits at their opposite ends [9]. A periodic array of SRRs was used for the first actual demonstration of a negative index refraction medium [10]. Other electrically small (subwavelength) particle elements inspired by the split ring resonator, such as the complementary split ring resonator (CSRR) [11], and the recently proposed use of open particles (i.e., the open split ring resonator (OSRR) and its complementary OCSRR) can be applied as well as the loading elements of a host transmission line [8]. The present chapter focuses on the automated synthesis of CSRR-based resonant-type artificial TLs, but there are works on the direct extension to other loaded lines, too.

The chapter is organized as follows. First, the CSRR-based artificial transmission lines are introduced and explained in detail. Starting from the physical topology of a

single unit cell, the EM behavior, its equivalent circuit, and the related characteristic electrical parameters are obtained in a straightforward way. From a practical point of view, this is very interesting, since this approach, which is based on SM techniques, greatly reduces the computational time related to the optimization process. Next, the implemented aggressive space mapping (ASM) technique is outlined, as well as its application to the automated synthesis of these lines. Finally, some examples illustrating the potential of this SM technique, along with the main conclusions and future directions in this area of research, are briefly discussed.

2 CSRR-Based Artificial Transmission Lines

In this type of artificial transmission line, the CSRRs usually appear etched on the metalized ground plane of the substrate, and the host line is typically implemented in microstrip technology. Other planar configurations are also possible. The first line under study has a simple geometrical topology that has been widely studied in the literature [12, 13]. ASM techniques have proven to be a suitable synthesis tool for these lines, and similar strategies can be extended to more complex topologies. The layout of a single unit cell is depicted in Fig. 1(a); it consists of a microstrip line with a CSRR etched on the ground plane just beneath the conductor strip. The geometrical dimensions are the width of the microstrip line W, the external CSRR radius r_{ext}, the width of the slots c, the distance between them d, and the split of the rings s_{split}. This structure presents a stop-band response, over a narrow band, as shown in Fig. 2(a). One of these unit cells, or some of the cells after cascading, can be practically used to implement a notch (or stop-band) filter [14, 15].

The second type of artificial transmission line considered is called a CSRR-gap-loaded line to make a clear distinction between it and the previous line. In this type, a simple series gap of length s is introduced in the conductor strip line. This gap is centered with respect to the CSRR locations, as illustrated in Fig. 1(b). Among other applications, there are practical applications of this type of artificial transmission line related to dual-band or enhanced bandwidth components. A slightly different version of this line which presents the series gap with a T-shaped geometry is also often used. This geometry allows an increase in the capacitance value for some practical applications [16, 17].

As the size of the loading elements, the CSRRs, is much smaller than the operating wavelength, the response of these two CSRR-based TLs can be accurately described by equivalent lumped circuit models [12, 13] in the region where these elements are operating. In Fig. 1, the equivalent model for each unit cell is presented as a reactive element, since the CSRR is modeled by the parallel tank composed of C_c and L_c. In the first case (Fig. 1(a)), L represents the line inductance and C the coupling capacitance between the line and the CSRR. We note that the ohmic losses of the CSRR could be easily taken into account by just including a parallel resistance in the model [12]. In the second case, L also represents the line inductance, while C and C_g model the gap and the coupling with the transmission line, respectively.

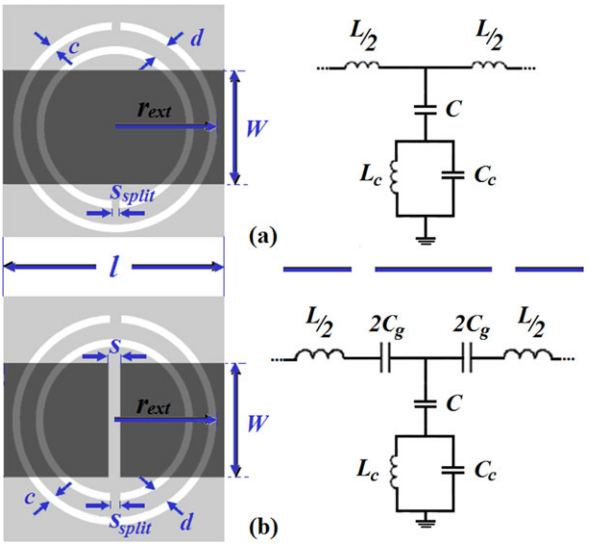


Fig. 1 Layout and equivalent circuit model of a unit cell of a CSRR-loaded line without (**a**) and with (**b**) series gap. The ground plane is depicted in *light gray*

This T-model is achieved after a transformation of the standard π-network [13] that models a series gap in a microstrip line [18]. The values of the circuit elements shown in Fig. 1 can be directly extracted from the EM simulation of these structures (see [12] for the details). The parameter extraction (PE) process is a crucial step in SM techniques, since it directly affects the convergence speed of these algorithms [19]. The agreement between the circuit simulation of the extracted values and the corresponding EM simulation is excellent, as illustrated with the two examples in Fig. 2. The process we have followed is briefly explained for completeness in the following paragraphs.

Two frequencies can be easily identified, by observing Figs. 1 and 2. The frequency that nulls the shunt impedance is represented by f_z, and the one that nulls the shunt admittance is represented by f_0:

$$f_z = \frac{1}{2\pi\sqrt{L_c(C+C_c)}}, \tag{1}$$

$$f_0 = \frac{1}{2\pi\sqrt{L_c C_c}}. \tag{2}$$

The representation of the magnitude of the transmission coefficient, i.e., S_{21}, allows us to identify clearly the transmission zero at f_z. On the other hand, the shunt branch becomes an open circuit for the value of f_0, which causes the reflection coefficient S_{11} to be intercepted with the unit resistance circle in the Smith chart. Moreover, at an angular frequency of $-\pi/2$ ($\pi/2$ for the CSRR-gap-loaded line case), the series and shunt impedances of the T-circuit model of the structure should be equal with opposite signs:

$$Z_s(\omega_{-\pi/2}) = -Z_p(\omega_{-\pi/2}). \tag{3}$$

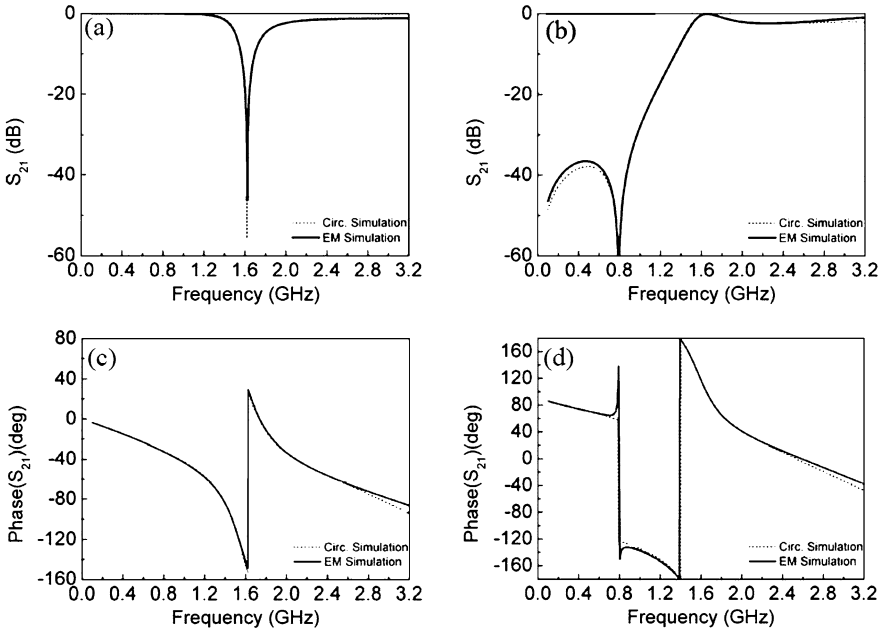


Fig. 2 Typical magnitude and phase of the transmission coefficient of the CSRR-based artificial transmission lines shown in Fig. 1. The plots (**a**) and (**c**) correspond to the CSRR-loaded line; (**b**) and (**d**) to the CSRR-gap-loaded line. EM and electrical simulations are depicted in *bold* and *dotted lines*, respectively

Hence, for the first line all the parameters are determined just by solving the equation system given by the previous expressions. In the CSRR-gap-loaded line, an additional condition is needed to fully determine the complete circuit, which has one more parameter (C_g). This extra condition is given by the resonance frequency of the series branch as follows:

$$f_s = \frac{1}{2\pi\sqrt{LC_g}}, \tag{4}$$

which is simply derived from the interception of S_{11} with the unit conductance circle, when the impedance on the series branch is rendered closed-circuit.

3 Application of SM Techniques

The main advantage of SM techniques is that they shift the optimization load from an expensive "fine" model to a cheaper "coarse" model. In the microwave area, fine models are often based on full-wave EM simulations, while coarse models usually employ equivalent circuits of the structures under study. After a reasonable number of fine model evaluations, SM algorithms lead in general to the construction of a

mapping between "fine" and "coarse" models providing a satisfactory solution in the accurate ("fine") simulation space.

The ASM method was originally introduced by Bandler et al. in 1995 [20]. It is one of the successive enhanced techniques that have been proposed in the literature since the appearance of the first SM approach [21]. In order to automate the synthesis procedure of CSRR-based artificial lines, the constrained version of the Broyden-based linear SM [22] is selected. Our approach uses constraints to avoid the instabilities of the Broyden-based linear SM algorithm, as well as to avoid reaching unwanted (from a practical realization point of view) solutions.

Each SM-based design strategy tries to find a mapping (P), which relates the design parameters of the EM model, denoted from now on as vector $\boldsymbol{x}_{\mathrm{em}}$, and the coarse model parameters, vector $\boldsymbol{x}_c$:

$$\boldsymbol{x}_c = P(\boldsymbol{x}_{\mathrm{em}}) \tag{5}$$

such that the different model responses, $R_{\mathrm{em}}(\boldsymbol{x}_{\mathrm{em}})$ and $R_c(\boldsymbol{x}_c)$, are aligned,

$$R_{\mathrm{em}}(\boldsymbol{x}_{\mathrm{em}}) \approx R_c\big(P(\boldsymbol{x}_{\mathrm{em}})\big), \tag{6}$$

in the region of interest. The solution $\boldsymbol{x}^*_{\mathrm{em}}$ is found via the inverse transformation of P:

$$\boldsymbol{x}^*_{\mathrm{em}} = P^{-1}\big(\boldsymbol{x}^*_c\big), \tag{7}$$

where $\boldsymbol{x}^*_c$ is the optimal coarse solution which provides the desired target response $R_c(\boldsymbol{x}^*_c)$. In the problems we considered, the coarse model parameters are the elements of the corresponding equivalent lumped circuit models. For the EM models, the dielectric characteristic of the substrate and its related thickness are known data, and the geometry dimensions are the design parameters. See Fig. 3 for each of the considered cases. The model response is related to the device behavior, so the scattering parameters are evaluated in a predefined frequency range.

Of course, the goal of ASM is to find the solution $\boldsymbol{x}_{\mathrm{em}}$ which approximates the optimal coarse model response $\boldsymbol{x}^*_c$. This means, expressed in mathematical terms, that we must solve the following set of nonlinear equations:

$$f(\boldsymbol{x}_{\mathrm{em}}) = P(\boldsymbol{x}_{\mathrm{em}}) - \boldsymbol{x}^*_c = 0, \tag{8}$$

where f is a suitable error function. Since ASM is an iterative algorithm, we will add a superscript to the notation in order to indicate the iteration number. Thus, $\boldsymbol{x}^{(j)}_{\mathrm{em}}$ indicates the j-th approximation to the solution layout. The next solution is predicted by applying a step $\boldsymbol{h}^{(j)}$ to the previous one in the quasi-Newton direction,

$$\boldsymbol{x}^{(j+1)}_{\mathrm{em}} = \boldsymbol{x}^{(j)}_{\mathrm{em}} + \boldsymbol{h}^{(j)}, \tag{9}$$

which is calculated as follows:

$$B^{(j)}h^{(j)} = -f^{(j)}, \tag{10}$$

where $\boldsymbol{B}^{(j)}$ is called the Broyden matrix [20, 22], which is properly updated in each iteration. Since the calculation of $\boldsymbol{h}^{(j)}$ requires the inverse of $\boldsymbol{B}^{(j)}$ according to (10), it is more convenient to have the same number of parameters in each model; as a

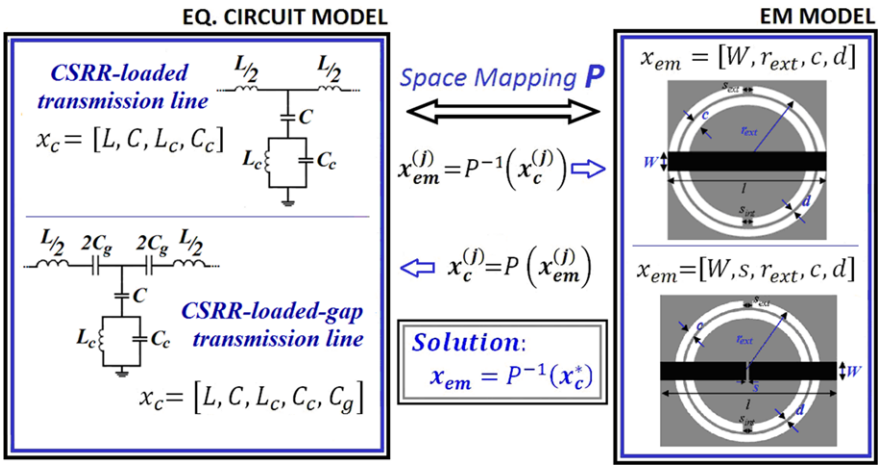


Fig. 3 Conceptual diagram of space mapping techniques for the two cells to be synthesized

result, $\boldsymbol{B}^{(j)}$ is a square, nonsingular matrix. Hence, the split of the CSRR s_{split} is considered a square gap of side c, and the length of the line l is fixed to the diameter value of the CSRR external ring. This is a reasonable assumption due to the fact that larger/smaller lengths of the line would lead to different equivalent circuit models, and on the other hand the CSRR sensitivity on the split size is not very high.

In the constrained approach, $\boldsymbol{h}^{(j)}$ is decreased in the same quasi-Newton direction by a shrinking factor δ, when the new solution $\boldsymbol{x}_{\text{em}}^{(j)}$ is not within the acceptable established limits. Dimensions that are too small may be impossible to implement due to technological limitations. Moreover, the equivalent circuit model used for the PE stage might be inaccurate for very extreme dimensions, which is a critical issue for ASM convergence. The algorithm continues with normal evolution once the quasi-Newton step is properly adjusted; see Fig. 4. Convergence is achieved when the norm of the error function $f^{(j)}$ given by:

$$\left\| f^{(j)} \right\| = \sqrt{\left(L^{(j)} - L^*\right)^2 + \left(C^{(j)} - C^*\right)^2 + \left(L_c^{(j)} - L_c^*\right)^2 + \left(C_c^{(j)} - C_c^*\right)^2} \quad (11)$$

is smaller than a fixed positive value $\eta \ll 1$. In the case of CSRR-gap-loaded lines the stopping criterion is slightly modified to $\| f_{\text{norm}}^{(j)} \| < \eta_1$ and/or $\| f^{(j)} \| < \eta_2$, where the normalized error function is defined as:

$$\left\| f_{\text{norm}}^{(j)} \right\| = \sqrt{\left(1 - \frac{L^{(j)}}{L^*}\right)^2 + \left(1 - \frac{C^{(j)}}{C^*}\right)^2 + \left(1 - \frac{L_c^{(j)}}{L_c^*}\right)^2 + \left(1 - \frac{C_c^{(j)}}{C_c^*}\right)^2} \quad (12)$$

and where η_1, η_2 are small positive numbers close to zero, $\eta_1 < \eta_2 \ll 1$.

For the algorithm initialization, given an ideal target frequency response characterized by $\boldsymbol{x}_c^*$, an initial layout $\boldsymbol{x}_{\text{em}}^{(1)}$ needs to be inferred. This is done by using well-known analytical formulas that link the electrical parameters with the geometrical dimensions of the host line and the CSRR resonator separately, thus not taking

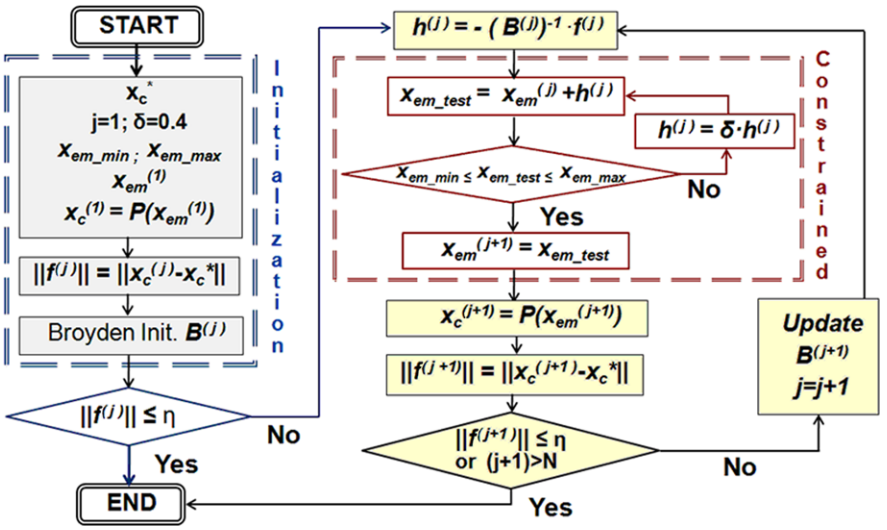


Fig. 4 Flow diagram of the proposed constrained ASM algorithm

into account the coupling between them. For the host line, different models are used depending on the presence or absence of the gap. In the simplest case (i.e., without the gap, see Fig. 1(a)), the characteristic impedance can be approximated by:

$$Z_0 = \sqrt{\frac{L_{\text{pul}}}{C_{\text{pul}}}} \approx \sqrt{\frac{L^*}{C^*}} \tag{13}$$

where L_{pul} and C_{pul} are the per-unit-length inductance and capacitance of the microstrip line, respectively. For a given impedance value and known dielectric constant ε_r, the ratio of the width W over the substrate height h can be calculated as:

$$\frac{W}{d} = \begin{cases} \frac{8e^A}{e^{2A}-2} & \text{for } (W/d) < 2, \\ \frac{2}{\pi}[B - 1 - \ln(2B-1) + \frac{\varepsilon_r - 1}{2\varepsilon_r}\{\ln(B-1) + 0.39 - \frac{0.61}{\varepsilon_r}\}] & \text{for } (W/d) > 2 \end{cases} \tag{14}$$

and hence the initial width is estimated [23]. The terms A and B are given by:

$$\begin{aligned} A &= \frac{Z_0}{60}\sqrt{\frac{\varepsilon_r + 1}{2}} + \frac{\varepsilon_r - 1}{\varepsilon_r + 1}\left(0.23 + \frac{0.11}{\varepsilon_r}\right), \\ B &= \frac{377\pi}{2Z_0\sqrt{\varepsilon_r}}. \end{aligned} \tag{15}$$

A host microstrip line with a gap, according to [18], can be modeled by means of a π-circuit composed of three capacitors (C_s for the series branch and C_p for the parallel capacitance). Those values, depicted in Fig. 5, can be connected with the ones of the π-circuit model in Fig. 1(b) through the following expressions:

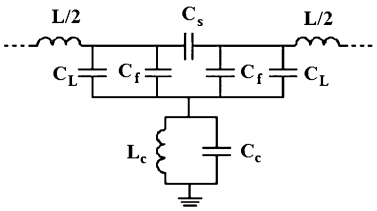


Fig. 5 Equivalent circuit to the simplified model of Fig. 1(b), proposed in [12]

$$C_p = \frac{2C_g^* C^*}{4C_g^* + C^*}, \tag{16}$$

$$C_s = \frac{(2C_g^*)^2}{4C_g^* + C^*}. \tag{17}$$

The parallel capacitance C_p is defined as:

$$C_p = C_f + C_L, \tag{18}$$

where C_L and C_f are the line and the fringing capacitance, respectively. Typically, C_p is dominated by the line capacitance; therefore, we can neglect the fringe capacitance in a first-order approximation, and consider that the parallel capacitance is equal to the line capacitance. Knowing C_L and L^*, the characteristic impedance of the host line is obtained following a similar procedure to (13), and consequently the initial width can also be determined for the line [23].

Now, in order to estimate the CSRR initial dimensions, we need to perform a very simple optimization process. Analytical formulas for the approximate characteristic electrical parameters of a given topology are known (see [24]). In this case, we fix one of the CSRR geometry parameters to a constant value (typically c) to ease the optimization process followed to find a deterministic solution for the initial layout. Therefore, we find the optimal values of r and d that minimize the error function between the electrical parameters of the real structure (L_c, C_c) and the target ones (L_c^*, C_c^*):

$$f_{\text{error}} = \frac{\sqrt{(L_c - L_c^*)^2 + (C_c - C_c^*)^2}}{\sqrt{(L_c^*)^2 + (C_c^*)^2}}. \tag{19}$$

The procedure followed for the initialization of the Broyden matrix $\boldsymbol{B}$ is based on a finite difference scheme. Although the identity matrix is typically chosen for initialization, a different approach was proposed in order to help the algorithm converge faster, considering that the design parameters are completely different in each model, i.e., physical versus electrical ones. To this extent, each geometrical parameter is slightly perturbed from the estimated initial value, the new layout is simulated, and then the corresponding electrical parameters are extracted. Thus, the initial Broyden matrix is calculated as follows:

$$\boldsymbol{B}^{(1)} = \begin{pmatrix} \partial L/\partial W & \partial L/\partial r_{\text{ext}} & \partial L/\partial c & \partial L/\partial d \\ \partial C/\partial W & \partial C/\partial r_{\text{ext}} & \partial C/\partial c & \partial C/\partial d \\ \partial L_c/\partial W & \partial L_c/\partial r_{\text{ext}} & \partial L_c/\partial c & \partial L_c/\partial d \\ \partial C_c/\partial W & \partial C_c/\partial r_{\text{ext}} & \partial C_c/\partial c & \partial C_c/\partial d \end{pmatrix}. \tag{20}$$

To conclude this section, some aspects regarding the practical implementation of the proposed SM method are discussed (further details can be found in [25]). The core of the program (the ASM algorithm) has been implemented in MATLAB [26]. External calls to a microwave engineering CAD suite that allows full-wave simulations are done using scripts. After each layout simulation, MATLAB recovers the control and makes use of the S-parameters to extract the electrical parameters in a straightforward way, as was previously explained. Next, the norm of the error function is evaluated (11)–(12). Normal ASM execution continues if convergence is not reached, or if the maximum number of EM simulations N (established by the user) is not exceeded; see Fig. 4. Thus, a fully automated tool [27–29] that makes it possible to synthesize these artificial transmission lines is available. As a consequence, the design process is shortened from many hours/days to several minutes/hours, and does not require the user to have special designing skills.

4 Validation Examples

In order to demonstrate the capabilities of the proposed synthesis technique, different examples (one for each cell of study) are presented. The optimal coarse solutions for both cases are taken from [12], with the aim of checking whether the proposed synthesis method provides final layouts close enough to the ones already given in [12]. The selected substrate is *Rogers RO3010* with a thickness of $h = 1.27$ mm, a dielectric constant of $\varepsilon_r = 10.2$, and loss tangent $\tan\delta = 0.0023$. Dielectric losses were not taken into account for the EM simulations, and metal parts were considered as perfect conductors (since we did not include a resistance in the equivalent circuit model). For each example we have made use of a different commercial EM solver: *Ansoft Designer* [30] was used in the first case and *Agilent Momentum* [31] in the second one.

4.1 Example of a CSRR-Loaded TL

The optimal target solution $\boldsymbol{x}_c^*$ (see Table 1) has a transmission zero at $f_z = 1$ GHz. The initial layout, $\boldsymbol{x}_{\text{em}}^{(1)}$, is quite shifted from the target response (see Fig. 7), but it can still be considered a good starting point for initiating the algorithm. Nevertheless, it is obvious that a better initial point (smaller value of $\|f^{(1)}\|$) would lead to a faster convergence of the algorithm. In Fig. 6 one can clearly see the evolution of the error function with the execution of the ASM algorithm; it becomes rather small after just ten iterations. The final synthesis was obtained in iteration 17, which means a CPU effort of approximately 30 min (using a standard computer with 3 GB of RAM and a 2.86 GHz clock processor).

A good matching for the responses of the optimal coarse solution $\boldsymbol{x}_c^*$ and the final optimum layout $\boldsymbol{x}_{\text{em}}^*$ can be clearly seen in Fig. 7. The dimensions of the final unit cell are summarized in Table 2, where the initial layout is also included.

Table 1 Coarse solutions and norm of the error function

	L [nH]	C [pF]	L_c [nH]	C_c [nH]	$\|f\|$
Target $\boldsymbol{x}_c^*$	5.080	4.430	2.980	4.060	
Initial $\boldsymbol{x}_c^{(1)}$	8.201	2.793	2.672	4.109	3.5377
Final $\boldsymbol{x}_c^{(17)}$	5.076	4.420	2.986	4.063	0.0124

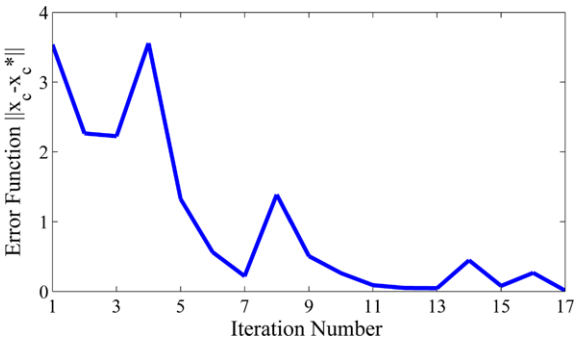


Fig. 6 Evolution of $\|f\|$ versus iteration number for the first example

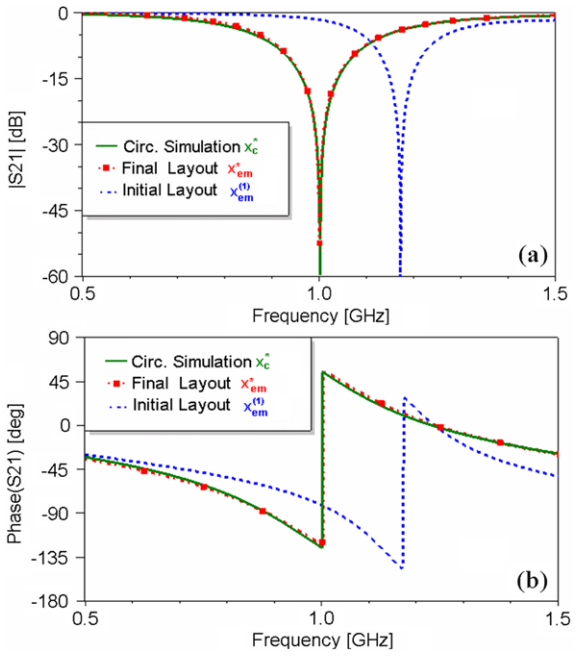


Fig. 7 Magnitude (**a**) and phase (**b**) of transmission coefficient S_{21} at initial solution $\boldsymbol{x}_{\text{em}}^{(1)}$, final solution $\boldsymbol{x}_{\text{em}}^*$, and target circuital solution $\boldsymbol{x}_c^*$

Table 2 Layout dimensions

	r_{ext} [mm]	c [mm]	d [mm]	W [mm]	l_{strip} [mm]
Initial $\boldsymbol{x}_{\text{em}}^{(1)}$	6.39	0.25	0.95	2.43	12.78
Final $\boldsymbol{x}_{\text{em}}^{*}$	5.67	0.33	0.34	4.93	11.34

Table 3 Layout dimensions

	r_{ext} [mm]	c [mm]	d [mm]	W [mm]	s [mm]
Initial $\boldsymbol{x}_{\text{em}}^{(1)}$	5.46	0.25	0.33	1.14	0.11
Final $\boldsymbol{x}_{\text{em}}^{*}$	5.62	0.31	0.19	3.90	0.31

4.2 Example of a CSRR-Gap-Loaded TL

The target parameters of the circuit model for this example are $L^* = 4.92$ nH, $C^* = 35.87$ pF, $L_c^* = 3.41$ nH, $C_c^* = 3.85$ pF, and $C_g^* = 1.05$ pF. Note that the requested value for the coupling capacitance C^* is big due to the presence of the gap, and since its magnitude is much larger than the rest of the circuit parameters, the norm of the normalized error (12) has been used for the stopping criterion. The final synthesis data (see Table 3) were obtained after 25 iterations, which means around 90 min of CPU time, and it can be seen that they are really close to the ones provided in [12] (which were derived via a manual optimization procedure).

The agreement between the circuit simulation of $\boldsymbol{x}_c^*$ and the final EM simulation $\boldsymbol{x}_{\text{em}}^*$ is indeed very good, as it can be see in Fig. 8. There is a slight discrepancy between the measurements and simulations attributed to the tolerances of the fabrication process.

5 Enhancements

In order to increase the robustness and accelerate the convergence rate of the algorithm, some other enhancements have been implemented. For instance, a line search (LS) technique has been introduced in the ASM update stage, with the aim of avoiding abrupt variations in the error norm evolution $\|f\|$ [32]. Introducing LS causes $\|f\|$ to follow a global decreasing behavior until convergence is achieved. See Fig. 9.

When the error function is enormously increased with respect to the previous iteration, taking the full Newton step $\boldsymbol{h}^{(j)}$ is a bad choice. The length of the step is controlled with LS in the same Newton direction by means of a factor λ:

$$\boldsymbol{x}_{\text{em}}^{(j+1)} = \boldsymbol{x}_{\text{em}}^{(j)} + \lambda \cdot \boldsymbol{h}^{(j)}, \quad 0 < \lambda \leq 1. \tag{21}$$

First, λ is chosen equal to unit (i.e., the normal evolution of the algorithm), and if the solution $\boldsymbol{x}_{\text{em}}^{(j+1)}$ does not meet the following criterion:

$$f\left(\boldsymbol{x}_{\text{em}}^{(j+1)}\right) \leq f\left(\boldsymbol{x}_{\text{em}}^{(j)}\right) + \kappa \cdot \nabla f\left(\boldsymbol{x}_{\text{em}}^{(j)}\right) \cdot \lambda \cdot \boldsymbol{h}^{(j)} \quad \text{with } \kappa = 10^{-4}, \tag{22}$$

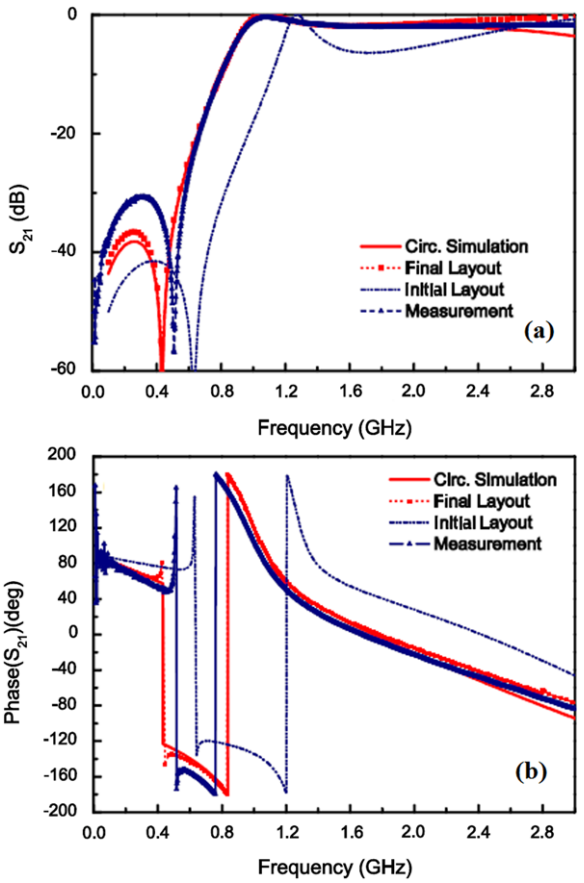


Fig. 8 Magnitude (**a**) and phase (**b**) of the transmission coefficient of the structure previously published in [29]. The initial solution $\boldsymbol{x}_{\text{em}}^{(1)}$, final solution $\boldsymbol{x}_{\text{em}}^{*}$, target response $\boldsymbol{x}_{c}^{*}$, and measurements are drawn with *blue dotted*, *red dotted*, *red solid*, and *solid blue lines*, respectively

several backtracks along the Newton directions are applied until the error function is decreased, or $\boldsymbol{x}_{\text{em}}^{(j+1)}$ is equal to $\boldsymbol{x}_{\text{em}}^{(j)}$. For the first backtrack, the error function is modeled by a quadratic polynomial function, and in the following ones by means of a cubic polynomial approach; see [33] for more details about the LS technique. One example where LS results in a clear improvement is the one shown in Fig. 10. The synthesized cell corresponds to the first type of lines (Fig. 1(a)) with a target response given by $L^{*}[\text{nH}] = 4.860$, $L_c[\text{nH}] = 2.180$, $C^{*}[\text{pF}] = 1.640$, and $C_c^{*}[\text{pF}] = 2.890$. The number of EM simulations needed to find the final synthesis is decreased from 29 to 18, hence reducing the computational time significantly. The initial layout and the corresponding final synthesis dimensions are summarized in Table 4. The ASM constrained approach used a shrinking factor of $\delta = 0.4$.

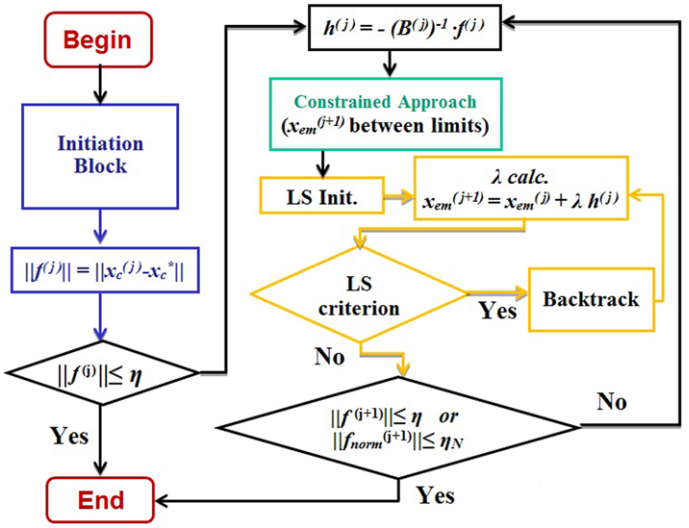


Fig. 9 Flow diagram of the ASM algorithm with the LS technique

Table 4 Layout dimensions and error function values

	r_{ext} [mm]	c [mm]	d [mm]	W [mm]	$\|f\|$
Initial $\boldsymbol{x}_{\text{em}}^{(1)}$	4.83	0.25	0.93	0.99	3.105
ASM $\boldsymbol{x}_{\text{em}}^{*}$ ($\delta = 0.4$)	4.02	0.42	0.15	1.95	0.048
ASM + LS $\boldsymbol{x}_{\text{em}}^{*}$ ($\delta = 0.4$)	4.03	0.43	0.16	1.99	0.050

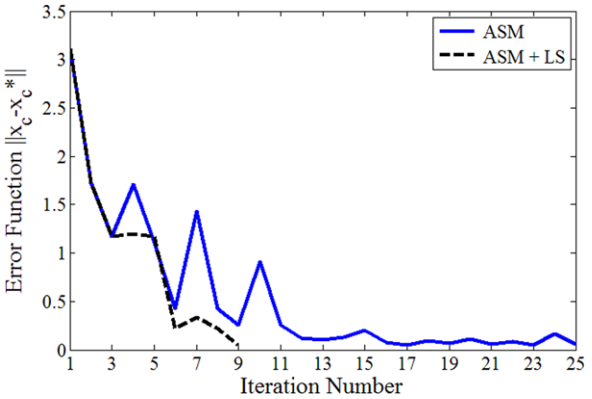


Fig. 10 Evolution of $\|f\|$ versus iteration number, using ASM with $\delta = 0.4$ (*solid blue line*) and ASM + LS (*dashed black line*)

After applying the LS algorithm to several examples, we have found that the introduction of LS does not always provide the expected positive effects. In some cases, we have observed that the convergence of the algorithm with LS can be-

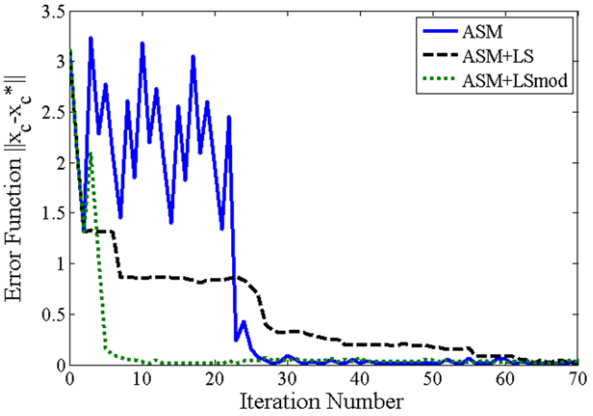


Fig. 11 Evolution of $\|f\|$ versus iteration number, using ASM (*solid blue line*), ASM + LS (*dashed black line*), and ASM + LS modified (*dotted green line*)

Table 5 Layout dimensions and error function values

	r_{ext} [mm]	c [mm]	d [mm]	W [mm]	$\|f\|$
Initial $\boldsymbol{x}_{\text{em}}^{(1)}$	4.83	0.25	0.93	0.99	3.1053
ASM $\boldsymbol{x}_{\text{em}}^{*}$ ($\delta = 0.3$)	4.02	0.40	0.18	1.93	0.0104
ASM + LS $\boldsymbol{x}_{\text{em}}^{*}$ ($\delta = 0.3$)	4.03	0.38	0.19	1.95	0.0137
ASM + LS mod $\boldsymbol{x}_{\text{em}}^{*}$ ($\delta = 0.3$)	4.02	0.40	0.18	1.93	0.0104

come slower than without it. Each backtrack performed implies an additional EM simulation. If the error is high (far from the final synthesis) it has no sense to perform several backtracks when previous iteration presents an error of the same order. Many backtracks will have a high cost and not a real benefit, it is enough to apply a small correction in order not to slow the progress of the algorithm. Therefore, only one backtrack is performed while the norm of the error (or normalized error for the case of CSRR-gap loaded lines) is still far from the stopping criterion for the ASM algorithm. In Fig. 11 this fact is illustrated using the same target response as tested previously but a different shrinking factor: $\delta = 0.3$. The algorithm is forced to continue until 70 iterations to clearly appreciate the softer evolution of the algorithm when standard LS is applied. For the modified LS algorithm, convergence is already achieved at iteration 13, whereas the normal execution of ASM ($\delta = 0.3$) converges at iteration 44. All solutions for this last example are collected in Table 5. It is remarkable that sometimes choosing a bad value of δ can also lead to a slow convergence of the algorithm, and this modified solution of LS can help to minimize that effect.

6 Conclusions

In conclusion, it has been demonstrated that SM techniques are very valuable when used for the automated and efficient synthesis of CSRR-based artificial transmission lines. A robust method, which provides very good agreement between the target response and the EM simulation of the final layout solution, has been successfully proposed. This work opens a promising path towards the application of similar methodologies to synthesize other artificial transmission lines, as well as more complex passive devices based on them.

References

1. Bandler, J.W., Cheng, Q.S., Dakroury, S.A., Mohamed, A.S., Bakr, M.H., Madsen, K., Sondergaard, J.: Space mapping: the state of the art. IEEE Trans. Microw. Theory Tech. (2004). doi:10.1109/TMTT.2003.820904
2. Gil Barba, M.: Resonant-type metamaterial transmission lines and their application to microwave device design. Ph.D. dissertation, Universitat Autònoma de Barcelona (2009)
3. Durán-Sindreu, M.: Miniaturization of planar microwave components based on semi-lumped elements and artificial transmission lines. Ph.D. dissertation, Universitat Autònoma de Barcelona (2011)
4. Eleftheriades, G.V.: EM transmission-line metamaterials. Mater. Today **12**, 30–41 (2009)
5. Sievenpiper, D., Zhang, L., Bross, R.F.J., Alexópolous, N.G., Yablonovitch, E.: High-impedance electromagnetic surface with a forbidden frequency band. IEEE Trans. Microw. Theory Tech. **47**(11), 2059–2074 (1999). doi:10.1109/22.798001
6. Marqués, R., Martín, F., Sorolla, M.: Metamaterials with Negative Parameters: Theory, Design and Microwave Applications. Wiley, New York (2007)
7. Caloz, C., Itoh, T.: Novel microwave devices and structures based on the transmission line approach of meta-materials. Paper presented at IEEE MTT International Microwave Symposium, Philadelphia, PA (USA) (June 2003). doi:10.1109/MWSYM.2003.1210914
8. Durán-Sindreu, M., Aznar, F., Vélez, A., Bonache, J., Martín, F.: Analysis and applications of OSRR- and OCSRR-loaded transmission lines: A new path for the design of compact transmission line metamaterials. Paper presented at 3rd International Congress on Advanced Electromagnetics Materials in Microwaves and Optics, London (UK), Sept. 2009. doi:10.1016/j.bbr.2011.03.031
9. Marqués, R., Martel, J., Mesa, F., Medina, F.: Left-handed-media simulation and transmission of EM waves in subwavelength split-ring-resonator-loaded metallic waveguides. Phys. Rev. Lett. **89**(18) (2002). doi:10.1103/PhysRevLett.89.183901
10. Pendry, J.B., Holden, A.J., Robbins, D.J., Stewart, W.J.: Magnetism from conductors and enhanced nonlinear phenomena. IEEE Trans. Microw. Theory Tech. **47**(11), 2075–2084 (1999). doi:10.1109/22.798002
11. Falcone, F., Lopetegi, T., Laso, M.A.G., Baena, J.D., Bonache, J., Beruete, M., Marqués, R., Martín, F., Sorolla, M.: Babinet principle applied to the design of metasurfaces and metamaterials. Phys. Rev. Lett. **93**(4) (2004). doi:10.1103/PhysRevLett.93.197401
12. Bonache, J., Gil, M., Gil, I., Garcia-García, J., Martín, F.: On the electrical characteristics of complementary metamaterial resonators. IEEE Microw. Wirel. Compon. Lett. **16**(10), 543–545 (2006). doi:10.1109/LMWC.2006.882400
13. Bonache, J., Gil, M., Gil, I., Garcia-García, J., Martín, F.: Parametric analysis of microstrip lines loaded with complementary split ring resonators. Microw. Opt. Technol. Lett. **50**, 2093–2096 (2008). doi:10.1002/mop.23571
14. Falcone, F., Lopetegi, T., Baena, J.D., Marqués, R., Martín, F., Sorolla, M.: Effective negative-ε stopband microstrip lines based on complementary split ring resonators. IEEE Microw. Wirel. Compon. Lett. **14**(6), 280–282 (2004). doi:10.1109/LMWC.2004.828029

15. Kshetrimayum, K.S., Kallapudi, S., Karthikeyan, S.S.: Stop band characteristics for periodic patterns of CSRRs in the ground plane. IETE Tech. Rev. **24**(6), 449–460 (2007)
16. Bonache, J., Siso, G., Gil, M., Martín, F.: Dispersion engineering in resonant type metamaterial transmission lines and applications. In: Zouhdi, S. (ed.) Metamaterials and Plasmonics: Fundamentals, Modelling, Applications, pp. 269–276. Springer, New York (2009)
17. Siso, G., Bonache, J., Martín, F.: Miniaturization and dual-band operation in planar microwave components by using resonant-type metamaterial transmission lines. Paper presented at IEEE MTT-S International Microwave Workshop Series (IMWS 2008). Chengdu (China). doi:10.1109/IMWS.2008.4782255
18. Gupta, K.C., Garg, R., Bahl, I., Bhartia, P.: Microstrip Lines and Slotlines, 2nd edn. Artech House Microwave Library (1996)
19. Bakr, M.H., Bandler, J.W., Georgieva, N., Madsen, K.: A hybrid aggressive space-mapping algorithm for EM optimization. IEEE Trans. Microw. Theory Tech. **47**(12), 2440–2449 (1999). doi:10.1109/22.808991
20. Bandler, J.W., Biernacki, R.M., Chen, S.H., Hemmers, R.H., Madsen, K.: Electromagnetic optimization exploiting aggressive space mapping. IEEE Trans. Microw. Theory Tech. **43**(12), 2874–2882 (1995). doi:10.1109/22.475649
21. Bandler, J.W., Biernacki, R.M., Chen, S.H., Grobelny, P.A., Hemmers, R.H.: Space mapping technique for electromagnetic optimization. IEEE Trans. Microw. Theory Tech. **42**(12), 2536–2544 (1994). doi:10.1109/22.339794
22. Rayas-Sanchez, J.E., Gutierrez-Ayala, V.: EM-based Monte Carlo analysis and yield prediction of microwave circuits using linear-input neural-output space mapping. IEEE Trans. Microw. Theory Tech. **54**(12), 4528–4537 (2006). doi:10.1109/TMTT.2006.885902
23. Pozar, D.M.: Microwave Engineering, 3rd edn. Wiley, New York (2005)
24. Baena, J.D., Bonache, J., Martín, F., Marqués, R., Falcone, F., Lopetegi, I., Laso, M.A.G., García, J., Gil, I., Flores, M., Sorolla, M.: Equivalent circuit models for split ring resonators and complementary split ring resonators coupled to planar transmission lines. IEEE Trans. Microw. Theory Tech. **53**(4), 1451–1461 (2005). doi:10.1109/TMTT.2005.845211
25. Selga, J., Rodríguez, A., Gil, M., Carbonell, J., Boria, V.E., Martín, F.: Synthesis of planar microwave circuits through aggressive space mapping using commercially available software packages. Int. J. RF Microw. Comput.-Aided Eng. **20**, 527–534 (2010). doi:10.1002/mmce.20458
26. MATLAB version 2010a, The MathWorks Inc., Natick, MA (2010)
27. Rodriguez, A., Selga, J., Gil, M., Carbonell, J., Boria, V.E., Martin, F.: Automated synthesis of resonant-type metamaterial transmission lines using aggressive space mapping. In: Microwave Symposium Digest (MTT), 2010 IEEE MTT-S International, 23–28 May, pp. 209–212 (2010). doi:10.1109/MWSYM.2010.5517444
28. Selga, J., Rodriguez, A., Gil, M., Carbonell, J., Boria, V.E., Martín, F.: Towards the automatic layout synthesis in resonant-type metamaterial transmission lines. IET Microw. Antennas Propag. **4**(8), 1007–1015 (2010). doi:10.1049/iet-map.2009.0551
29. Selga, J., Rodriguez, A., Boria, V.E., Martin, F.: Application of aggressive space mapping to the synthesis of composite right/left handed (CRLH) transmission lines based on complementary split ring resonators (CSRRs). In: 2011 41st European Microwave Conference (EuMC), 10–13 Oct., pp. 968–971 (2011)
30. Ansoft Designer version 6.1.0. Ansys Inc., Canonsburg, PA (2010)
31. Agilent Momentum version 8.20.374. Agilent Technologies, Santa Clara, CA (2008)
32. Rodríguez, A., Selga, J., Martín, F., Boria, V.E.: On the implementation of a robust algorithm which automates the synthesis of artificial transmission lines based on CSRRs. In: Proceedings of International Congress on Advanced Electromagnetic Materials in Microwaves and Optics, Barcelona, Spain, Oct. (2011)
33. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C: The Art of Scientific Computing, 2nd edn. Cambridge University Press, Cambridge (1992)

The Efficiency of Difference Mapping in Space Mapping-Based Optimization

Murat Simsek and Neslihan Serap Sengor

Abstract In space mapping, a time-consuming but accurate fine model is used along with a less accurate but fast coarse model to reduce the overall computational effort. In this work, techniques using the difference mapping concept will be introduced. These techniques are efficient in reducing the computational effort while improving convergence. Difference mapping is constructed similarly to the mechanism used in space mapping, but, unlike space mapping, it facilitates the use of terminating conditions based on the simultaneous use of input and output values. Rigorous mathematical expressions related to difference mapping techniques will be given, and the improvement provided by these techniques will be discussed. Furthermore, to expose the efficiency of using the difference in input and output, simulation results obtained for high-dimensional applications will be given.

Keywords Artificial neural network · Knowledge-based modeling · Difference mapping · Inverse scattering · Optimization

1 Introduction

The efficiency of a design process is mainly improved through design optimization, in which appropriate values for design variables are determined. Conventional optimization techniques involve long iterative processes to obtain satisfactory results. In cases where expensive execution efforts are needed for a response, design optimization does not give satisfactory results due to time limitations.

Surrogate-based optimization techniques [3, 9, 21] have emerged to eliminate expensive, time-consuming calculations during the conventional optimization process. In these techniques, more accurate but time-consuming fine model execution

M. Simsek (✉)
Faculty of Aeronautics and Astronautics, Istanbul Technical University, Istanbul, Turkey
e-mail: simsekmu@itu.edu.tr

N.S. Sengor
Electronics and Communication Engineering, Istanbul Technical University, Istanbul, Turkey
e-mail: sengorn@itu.edu.tr

S. Koziel, L. Leifsson (eds.), *Surrogate-Based Modeling and Optimization*,
DOI 10.1007/978-1-4614-7551-4_5,

is utilized in a limited number of iterations. Less accurate but cheaper coarse model execution is mainly utilized during the optimization process; hence, the time spent on the design process is decreased using coarse model optimization instead of fine model optimization.

The space mapping-based optimization technique was first introduced by Bandler in 1994 [5] as an easily applicable surrogate-based optimization technique. Space mapping (SM) is based on constructing parameter mapping from the fine model parameter space to the coarse model parameter space. With this mapping between parameter spaces the coarse model becomes more related to the fine model and it provides approximate responses more similar to the fine model responses. SM exploits affine mapping to construct a mathematical link between the parameter spaces of coarse and fine models [4, 13].

In order to enhance SM-based optimization, the aggressive space mapping (ASM) technique was developed [6, 7]. The iterative method in ASM, namely the quasi-Newton method, is used to update the mapping while the mapping in SM is updated using the approximate matrix inverse. In ASM, a projection of the error between the responses of the fine and coarse models is used to determine the new iteration point. This projection could be built by using Broyden's update, which results in better convergence performance for ASM.

The difference mapping idea was introduced to exploit the use of existing knowledge obtained from the fine model such as a mathematical formulation, a simple equivalent model, or an artificial neural network model which would be based on the input–output relationship of the fine model. This a priori knowledge is known as the coarse model in surrogate-based techniques. The knowledge-based techniques [22] also use a priori knowledge, especially during the modeling process. Prior knowledge input with difference (PKI-D) [16, 17, 20] is developed especially to make use of existing knowledge twice. Therefore, using both input and output knowledge it is shown that it is possible to get more accurate results through mappings using input and output relationships. Difference mapping is based on PKI-D modeling; it involves embedding existing knowledge into a mapping where the output is the difference between fine and coarse model parameters. Since this technique depends on forming a mapping based on the difference, it is called difference mapping to distinguish it from other SM-based techniques.

Through a combination of difference mapping and the inverse coarse model obtained from the fine model using the relation between its input and outputs, the space mapping with difference (SM-D) technique [15, 18] has been developed. Moreover, to eliminate the inverse matrix calculation, the space mapping with inverse difference (SM-ID) technique [18, 19] has been developed by combining inverse difference mapping and the inverse coarse model. Both techniques show fast convergence, as they are based on a limited number of fine model executions. As a priori knowledge from the fine model is used, more accurate results are obtained, as shown by the lower error value compared with those of similar methods [8, 10, 14, 19].

2 Fundamentals of Space Mapping Optimization

Space mapping depends on using a coarse model in an efficient manner during optimization instead of using the fine model. Fine model execution is only needed to correct the mapping formed between the fine and coarse model parameters during optimization. In the SM technique, a more accurate fine model (f_f) with a higher computational burden is used with a less accurate but computationally cheap coarse model (f_c) via mapping $\mathbf{P}(\cdot)$ during iterations. Consequently, the SM optimization technique combines the computational efficiency of the coarse model with the accuracy of the fine model, giving rise to a technique that is both computationally efficient and accurate. $\mathbf{P}(\cdot)$ constructs a mapping from the fine model parameters to the coarse model parameters. The purpose of the SM technique is to obtain a desired input design parameter $\overline{\mathbf{x}}_f$ set which satisfies the target fine model response $\overline{\mathbf{Y}}_f$. The original optimization problem can be represented as follows:

$$\overline{\mathbf{x}}_f = \min_{\mathbf{x}} \left\| \overline{\mathbf{Y}}_f - \mathbf{f}_f(\mathbf{x}) \right\|. \tag{1}$$

This problem can be solved using direct optimization, but as using the fine model is a time-consuming and complex process, a surrogate-based optimization technique, especially SM, is preferred to obtain the desired input parameters of the fine model. During the construction of the mapping between the fine and coarse model parameter spaces, a parameter extraction process [2, 7] is necessary. The parameter extraction process allows us to find the coarse model parameters corresponding to the fine model parameters, so that both models give the same response. Uniqueness is the biggest issue for parameter extraction. The general form of parameter extraction for iteration j is represented as follows:

$$\mathbf{x}_c^{(j)} = \min_{\mathbf{x}} \left\| \mathbf{Y}_f^{(j)} - \mathbf{f}_c(\mathbf{x}) \right\|. \tag{2}$$

The SM optimization mapping structure can be represented by

$$\mathbf{x}_c = \mathbf{P}(\mathbf{x}_f) = \mathbf{B}\mathbf{x}_f + \mathbf{c}, \tag{3}$$

$$\mathbf{P} = \left[[\mathbf{c}]_{n\times 1} \; [\mathbf{B}]_{n\times n} \right]_{n\times(n+1)}, \tag{4}$$

where $\mathbf{x}_f$, $\mathbf{x}_c$, $\mathbf{c}$, and $\mathbf{B}$ are the fine and coarse model design parameters, and the shifting and linear transformation terms of affine mapping $\mathbf{P}$, respectively.

The mapping $\mathbf{P}$ is constructed iteratively, and execution of the fine model is required for each step. The $\mathbf{Q}$ and $\mathbf{D}$ matrices given below are used to construct a new mapping $\mathbf{P}$ for iteration step j as in (7).

$$\mathbf{Q}^{(j)} = \left[\mathbf{x}_c^{(1)} \quad \ldots \quad \mathbf{x}_c^{(j)} \right]_{n\times j}, \tag{5}$$

$$\mathbf{D}^{(j)} = \begin{bmatrix} 1 & \ldots & 1 \\ \mathbf{x}_f{}^{(1)} & \ldots & \mathbf{x}_f^{(j)} \end{bmatrix}_{(n+1)\times j}, \tag{6}$$

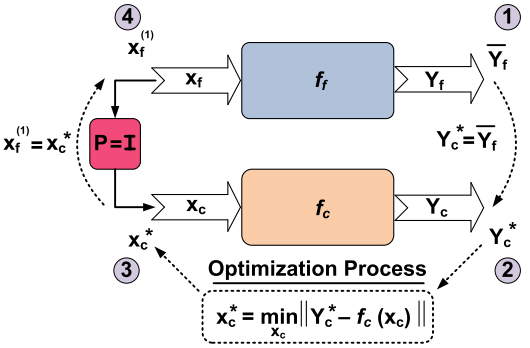


Fig. 1 Pre-iteration steps of space mapping technique

$$\mathbf{P}^{(j)} = \mathbf{Q}^{(j)}\mathbf{D}^{(j)^{\dagger}}. \tag{7}$$

The dimension of the matrix $\mathbf{D}$ is $(n+1) \times j$. If j is not equal to $n+1$, the inverse of non-square matrix $\mathbf{D}$ can be calculated using a pseudoinverse denoted by (†) as long as the full rank condition is satisfied. When the inverse of the non-square matrix is involved, the pseudoinverse technique gives an approximate solution in the sense of the least squares error.

After construction of the mapping using the fine and coarse model input parameters, a new fine model parameter is obtained by the inverse of mapping $\mathbf{P}$. This relation can be represented by

$$\mathbf{x}_f^{(j+1)} = \mathbf{P}^{(j)^{-1}}(\mathbf{x}_c^*), \tag{8}$$

where $\mathbf{P}^{(j)^{-1}}$ can be calculated by the inverse matrix formulation if the dimensions of the fine model parameter $\mathbf{x}_f$ and coarse model parameter $\mathbf{x}_c$ are the same. $\mathbf{x}_c^*$ in (8) is the optimum coarse model parameter satisfying the target fine model response for the coarse model. Before beginning iteration, several pre-iteration steps (see Fig. 1) should be completed as follows to determine the initial parameter values:

Step 1: Select $\overline{\mathbf{Y}}_f$ as an optimization point.
Step 2: Choose $\mathbf{Y}_c^* = \overline{\mathbf{Y}}_f$.
Step 3: Find $\mathbf{x}_c^*$ from $\mathbf{x}_c^* = \min_{\mathbf{x}_c} \|\overline{\mathbf{Y}}_f - \mathbf{f}_c(\mathbf{x}_c)\|$.
Step 4: Set $\mathbf{x}_f^{(1)} = \mathbf{x}_c^*$ and $j = 1$.

After the pre-iteration steps, the main iterations run until the stopping criterion is satisfied. The main iteration circle of the SM technique is given in Fig. 2. Each step of the SM algorithm is given below.

Step 1—Stopping Criterion:
calculate $\mathbf{Y}_f^{(j)} = \mathbf{f}_f(\mathbf{x}_f^{(j)})$ and if $\|\mathbf{Y}_f^{(j)} - \mathbf{Y}_c^*\| \leq \varepsilon$
then $\overline{\mathbf{x}}_f = \mathbf{x}_f^{(j)}$ else go to step 2.

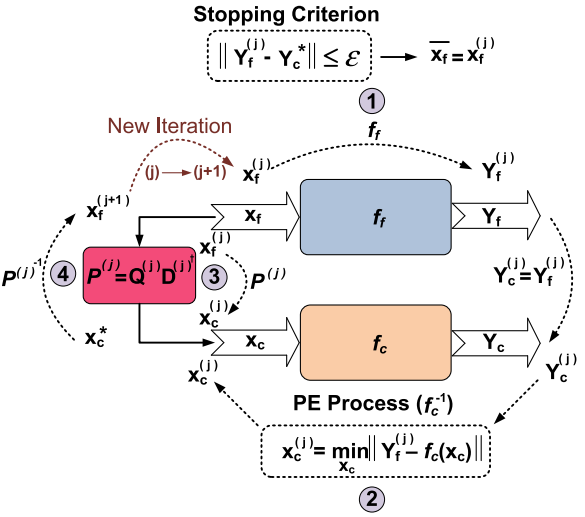


Fig. 2 Main iteration steps of space mapping technique

Step 2—Parameter Extraction:
find $\mathbf{x}_c^{(j)}$ using $\mathbf{x}_c^{(j)} = \min_{\mathbf{x}_c} \|\mathbf{Y}_f{}^{(j)} - \mathbf{f}_c(\mathbf{x}_c)\|$
Step 3—Forming P:
form $\mathbf{x}^{(j)} = \mathbf{Q}^{(j)}\mathbf{D}^{(j)\dagger}$
Step 4—New Iteration Point:
$\mathbf{x}_f^{(j+1)} = \mathbf{P}^{(j)\dagger}(\mathbf{x}_c^*)$, set $j = j + 1$ go to step 1.

The **Q** and **D** matrices are augmented at the end of each iteration step, so their dimensions are changed. This augmentation is due to the new knowledge added to these matrices to construct a suitable mapping during the iteration process. Therefore, at each new iteration point more information based on the relationship between the fine model parameters and corresponding coarse model parameters is used. The **D** and **Q** matrices are updated with new $\mathbf{x}_f$ and $\mathbf{x}_c$, respectively.

3 Difference Mapping Concept

The difference mapping concept has emerged with the idea of exploiting knowledge during optimization. Fine model execution is only used for parameter extraction in SM-based techniques; whereas, difference mapping makes use of the already-calculated fine model response besides the fine model parameters to construct the mapping. The output of this mapping is the difference between the fine and coarse model parameters. The fine model parameters are expected to be sufficiently close to the corresponding coarse model parameters. Using the difference between the fine and coarse model parameters restricts the range at the output of the difference mapping.

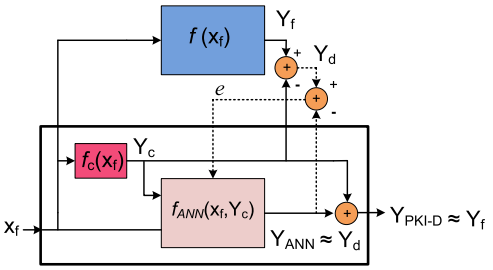


Fig. 3 Prior knowledge input with difference model structure [20]

To clarify the understanding of this new concept, the knowledge-based technique PKI-D can be considered. PKI-D is the latest knowledge-based technique and the most commonly used to improve the modeling efficiency by using coarse models. For the coarse model, an artificial neural network (ANN) structure is preferred, as its construction depends only on the input and output data, which can even be obtained from measurements. So there is no need for extra knowledge other than input-output relation. Thus, the coarse model response provides extra knowledge besides normal inputs to reduce the complexity of ANN mapping. As it is constructed to give the difference between the fine model response and the coarse model response, using the coarse model response twice gives a more accurate and less complex structure for ANN modeling. The mathematical formulation of the mapping structure for the PKI-D technique can be represented by

$$\mathbf{Y}_f \approx \mathbf{Y}_{\text{PKI-D}} = \mathbf{Y}_c + \underbrace{\mathbf{f}_{\text{ANN}}(\mathbf{x}_f, \mathbf{Y}_c)}_{\mathbf{Y}_d}, \tag{9}$$

where $\mathbf{Y}_{\text{PKI-D}}$ is the final model responses obtained from summation of the coarse model and ANN responses. $\mathbf{f}_{\text{ANN}}$ is ANN model with extra inputs to obtain the difference $\mathbf{Y}_d$ between $\mathbf{Y}_f$ and $\mathbf{Y}_c$, as shown in Fig. 3.

The key idea behind difference mapping is using two different information items in order to obtain efficient convergence to the desired fine model parameter. The first piece of information used is the coarse model parameters obtained from the coarse model through the parameter extraction process. SM-based techniques utilize this information to construct mappings between fine and coarse model parameters. The new iteration point of the fine model parameter is obtained by using the optimum coarse model parameter corresponding to the desired fine model parameters. With each iteration, more knowledge is provided to improve the mapping ability and the new iteration point converges to the desired fine model parameter more than the previous one.

Difference mapping utilizes fine model responses without an extra computational burden. This existing knowledge is embedded in the mapping structure; hence, the mapping constructed builds the connection between the optimum coarse model parameter and the desired fine model parameter. This connection, as it depends on the

fine model responses, provides extra knowledge during generation of the new iteration point. The new iteration point converges to the desired fine model parameter by satisfying two conditions simultaneously. The parameter-based and response-based conditions increase the efficiency of the convergence as they satisfy the following conditions:

$$\left\| \underbrace{\mathbf{P}\left(\mathbf{x}_f^{(j)}\right)}_{\mathbf{x}_c^{(j)}} - \mathbf{x}_c^* \right\| \leq \varepsilon_p, \qquad \left\| \mathbf{f}_c\left(\underbrace{\mathbf{P}\left(\mathbf{x}_f^{(j)}\right)}_{\mathbf{x}_c^{(j)}}\right) - \mathbf{Y}_c^* \right\| \leq \varepsilon. \tag{10}$$

Here, $\mathbf{x}_c^*$ and $\mathbf{Y}_c^*$ indicate the optimum coarse model parameter and optimum coarse model response, respectively. Thus, the constructed difference mapping satisfies these two conditions simultaneously, whereas other SM-based techniques satisfy only one.

4 Difference Mapping-Based Optimization Techniques

Using the difference mapping concept, two optimization techniques have been developed: space mapping with difference (SM-D) and space mapping with inverse difference (SM-ID). The main distinction between these two techniques is that in SM-ID, because it is based on inverse mapping, calculating an inverse is unnecessary. Since SM-ID is a new technique, this section will focus on it rather than SM-D. More information about SM-D can be obtained from [15, 18]. SM-ID involves inverse coarse model usage to eliminate the parameter extraction process. The inverse coarse model can be constructed by using an ANN. The considered ANN can be trained either with a small data set obtained from the fine model or with a larger data set obtained from the coarse model. The model parameters and model responses are used as inputs and outputs during the training process of a feedforward multi-layer perceptron. As the amount of training data increases, the accuracy of the inverse coarse model is increased. One hidden layer would be sufficient for the ANN, but two hidden layers are very convenient for solving highly nonlinear complex modeling problems. After the training process is completed, the inverse coarse model can generate coarse model parameters to construct difference mapping.

The SM-ID technique exploits inverse difference mapping ${}^{i}P_d^{(j)}$. The $\mathbf{Q}$ and $\mathbf{D}$ matrices are rearranged differently than in SM-based mapping, and they are given as follows:

$$\mathbf{Q}^{(j)} = \left[\underbrace{\mathbf{x}_f^{(1)} - \mathbf{x}_c^{(1)}}_{\mathbf{x}_d^{(1)}} \quad \underbrace{\mathbf{x}_f^{(2)} - \mathbf{x}_c^{(2)}}_{\mathbf{x}_d^{(2)}} \quad \cdots \quad \underbrace{\mathbf{x}_f^{(m_j)} - \mathbf{x}_c^{(m_j)}}_{\mathbf{x}_d^{(m_j)}} \right]_{n \times m_j}, \tag{11}$$

$$\mathbf{D}^{(j)} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \mathbf{x}_c^{(1)} & \mathbf{x}_c^{(2)} & \cdots & \mathbf{x}_c^{(m_j)} \\ \mathbf{Y}_f^{(1)} & \mathbf{Y}_f^{(2)} & \cdots & \mathbf{Y}_f^{(m_j)} \end{bmatrix}_{(1+n+m) \times m_j}. \tag{12}$$

The difference between the fine and coarse model parameters is represented as in (11). The structure of the **Q** and **D** matrices shows that the difference between fine and coarse model parameters $\mathbf{x}_f^{(j)} - \mathbf{x}_c^{(j)}$ is related not only to the coarse model parameter $\mathbf{x}_c^{(j)}$ but also to the fine model response $\mathbf{Y}_f^{(j)}$. The convergence and robustness of this technique against divergence improve due to this relation built into inverse difference mapping:

$$\mathbf{Q}^{(j)} = {}^{i}\mathbf{P}_d^{(j)}\mathbf{D}^{(j)}. \tag{13}$$

${}^{i}\mathbf{P}_d^{(j)}$ denotes the mapping from $\mathbf{D}^{(j)}$ to $\mathbf{Q}^{(j)}$ as shown in (13). The inverse difference mapping ${}^{i}\mathbf{P}_d^{(j)}$ is partitioned into two parts such as input mapping ${}^{i}\mathbf{P}_{d\text{in}}^{(j)}$ and output mapping ${}^{i}\mathbf{P}_{d\text{out}}^{(j)}$. The contributions of the coarse model parameter and fine model response are denoted by ${}^{i}\mathbf{P}_{d\text{in}}^{(j)}$ and ${}^{i}\mathbf{P}_{d\text{out}}^{(j)}$ in (14):

$$\begin{bmatrix} \mathbf{x}_d^{(1)} & \mathbf{x}_d^{(2)} & \cdots & \mathbf{x}_d^{(j)} \end{bmatrix} = \begin{bmatrix} [{}^{i}\mathbf{P}_{d\text{in}}^{(j)}] & [{}^{i}\mathbf{P}_{d\text{out}}^{(j)}] \end{bmatrix} \begin{bmatrix} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \mathbf{x}_c^{(1)} & \mathbf{x}_c^{(2)} & \cdots & \mathbf{x}_c^{(j)} \end{bmatrix} \\ \begin{bmatrix} \mathbf{Y}_f^{(1)} & \mathbf{Y}_f^{(2)} & \cdots & \mathbf{Y}_f^{(j)} \end{bmatrix} \end{bmatrix}. \tag{14}$$

After the inverse difference mapping is updated using (15), a new iteration point as the fine model parameter is generated by (16). Each data pair added to the inverse difference mapping improves the approximate iteration point, and it converges to the desired fine model parameter. The new iteration point is obtained by using the optimum coarse model parameter and optimum coarse model response, simultaneously. A simple form of (16) is given in (17).

$${}^{i}\mathbf{P}_d^{(j)} = \begin{bmatrix} [{}^{i}\mathbf{P}_{d\text{in}}^{(j)}] & [{}^{i}\mathbf{P}_{d\text{out}}^{(j)}] \end{bmatrix} = \mathbf{Q}^{(j)}\mathbf{D}^{(j)^{\dagger}} \tag{15}$$

$$\left[\mathbf{x}_f^{(j+1)} - \mathbf{x}_c^{*}\right] = \begin{bmatrix} [{}^{i}\mathbf{P}_{d\text{in}}^{(j)}] & [{}^{i}\mathbf{P}_{d\text{out}}^{(j)}] \end{bmatrix} \begin{bmatrix} \begin{bmatrix} 1 \\ \mathbf{x}_c^{*} \end{bmatrix} \\ \left[\mathbf{Y}_c^{*}\right] \end{bmatrix} \tag{16}$$

$$\left[\mathbf{x}_f^{(j+1)}\right] = {}^{i}\mathbf{P}_d^{(j)}\left(\mathbf{x}_c^{*}, \mathbf{Y}_c^{*}\right) + \left[\mathbf{x}_c^{*}\right]. \tag{17}$$

This inverse structure in Fig. 4 is constituted using feedforward ANN, which is convenient if it is not possible to get a mathematical formulation for the modeling. After the ANN training process, the third pre-iteration step is slightly different from that in SM. The rest of the pre-iteration process in SM-ID is the same as in SM except for the third step. Thus, only step 3 is given here:

Step 3: Find $\mathbf{x}_c^{*}$ from $\mathbf{x}_c^{*} = {}^{i}f_c(\mathbf{Y}_c^{*})$.

The main iteration steps of the SM-ID technique are denoted in Fig. 4. The SM-ID algorithm is given below:

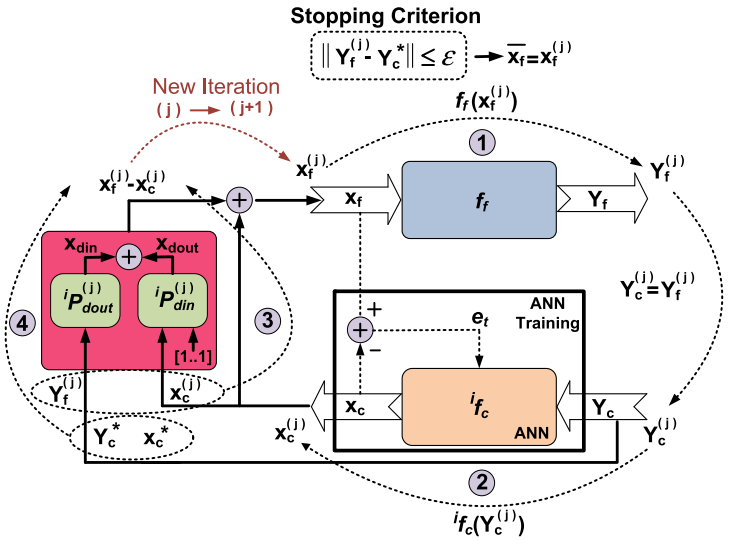


Fig. 4 Main iteration steps of space mapping with inverse difference (SM-ID) technique

Step 1—Stopping Criterion:
calculate $\mathbf{Y}_f^{(j)} = \mathbf{f}_f(\mathbf{x}_f^{(j)})$ and if $\|\mathbf{Y}_f^{(j)} - \mathbf{Y}_c^*\| \leq \varepsilon$
then $\overline{\mathbf{x}}_f = \mathbf{x}_f^{(j)}$ else go to step 2.
Step 2—Inverse Coarse Model Response:
find $\mathbf{x}_c^{(j)}$ using $\mathbf{x}_c^{(j)} = {}^i\mathbf{f}_c(\mathbf{Y}_c^{(j)})$
Step 3—Forming ${}^i\mathbf{P}_d$:
form ${}^i\mathbf{P}_d^{(j)} = \left[[{}^i\mathbf{P}_{d\text{in}}^{(j)}] \, [{}^i\mathbf{P}_{d\text{out}}^{(j)}] \right] = \mathbf{Q}^{(j)}\mathbf{D}^{(j)\dagger}$

$$\mathbf{Q}^{(j)} = \left[(\mathbf{x}_f^{(1)} - \mathbf{x}_c^{(1)}) \ldots (\mathbf{x}_f^{(j)} - \mathbf{x}_c^{(j)}) \right]_{n \times j}$$

$$\mathbf{D}^{(j)} = \begin{bmatrix} 1 & \ldots & 1 \\ \mathbf{x}_c^{(1)} & \ldots & \mathbf{x}_c^{(j)} \\ \mathbf{Y}_f^{(1)} & \ldots & \mathbf{Y}_f^{(j)} \end{bmatrix}^T_{(n+k+1) \times j}$$

where k denotes the number of coarse model outputs.
Step 4—New Iteration Point:
$\mathbf{x}_f^{(j+1)} = {}^i\mathbf{P}_{d\text{in}}^{(j)}(1, \mathbf{x}_c^*) + {}^i\mathbf{P}_{d\text{out}}^{(j)}(\mathbf{Y}_c^*) + \mathbf{x}_c^*$,
set $j = j + 1$ go to step 1.

5 An Application for Optimization

In this section, the convergence of the difference mapping-based technique SM-ID is discussed by comparing it with the SM-based technique ASM. Since ASM is a well-known technique and easy to apply to optimization problems, it is preferred

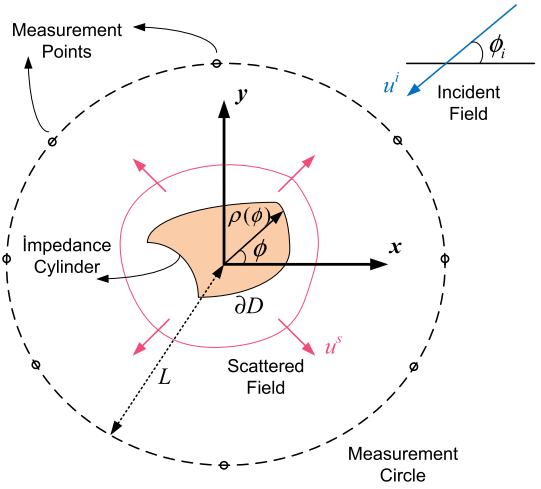


Fig. 5 Scattering problem geometry [19]

for comparison to demonstrate the convergence efficiency of SM-ID. The problem considered is a high inverse scattering problem. In order to show the effect of the coarse model, different inverse coarse models are generated using the ANN structure. These inverse coarse models are used for the parameter extraction process during the SM-ID and ASM optimization process.

5.1 Scattering Problem Definition

We consider the direct scattering problem depicted in Fig. 5. The arbitrarily shaped, infinitely long impedance cylinder in free space is illuminated by a plane wave whose polarization is the cylinder axis (z axis). The cylinder contour can be expressed by means of a Fourier series as

$$\rho(\phi) = \sum_{p=-P}^{P} a_p e^{ip\phi}, \qquad a_{-p} = a_p^*, \tag{18}$$

where a_p is the Fourier coefficient satisfying $a_{-p} = a_p^*$ and obtained as in (19),

$$a_p = \frac{1}{2\pi} \int_0^{2\Pi} \rho(\phi) e^{-ip\phi}\, d\phi. \tag{19}$$

The fine model constitutes a relation between the Fourier coefficients and the measured electric field, as depicted in Fig. 6. The formulation to obtain this relation is given here for the general case, and once the potential density φ is determined, the scattered field on the measurement points is obtained.

The electric field of the z-polarized incident wave can be expressed as

$$u^i(\mathbf{r}) = e^{-ik\hat{k}.\mathbf{r}}, \tag{20}$$

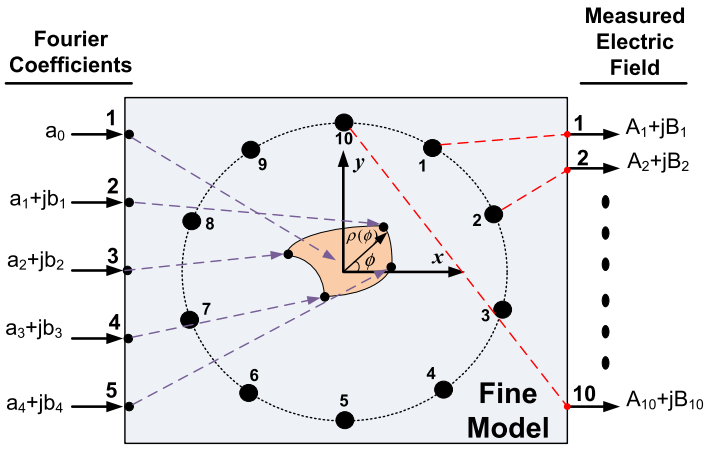


Fig. 6 Fine model for scattering problem

where $\mathbf{r} = x\hat{u}_x + y\hat{u}_y$ is a position vector and $\hat{k} = -(\sin\phi_i \hat{u}_x + \cos\phi_i \hat{u}_y)$ is a propagation unit vector of the incident wave with incident angle ϕ_i, and $k = \omega\sqrt{\varepsilon\mu}$ is the wave number of the background medium. Due to the homogeneity of the impedance cylinder with respect to the z axis, the total and the scattered electric fields also will be polarized parallel to the z axis. Therefore, the problem reduces to a transverse magnetic wave at direction z (TMz) case scalar problem in the x–y plane. The total field satisfies the impedance boundary condition [1] given in (21),

$$u(\mathbf{r}) + \frac{\eta}{ik}\frac{\partial u(\mathbf{r})}{\partial n} = 0, \quad \mathbf{r} \in \partial D, \tag{21}$$

where η is the normalized surface impedance of the boundary. As a conducting cylinder is considered, η is assumed to be almost zero. In order to obtained the scattered field, (22) must be solved, where the scattered field is expressed by a single layer potential as in [12],

$$u^s(\mathbf{r}) = \int_{\partial D} \varphi(\mathbf{r}')\Phi(\mathbf{r}, \mathbf{r}')\, ds(\mathbf{r}'), \quad \mathbf{r} \in R^2/\bar{D}, \tag{22}$$

where $\Phi(\mathbf{r}, \mathbf{r}')$ is Green's function in two dimensions given as

$$\Phi(\mathbf{r}, \mathbf{r}') = \frac{i}{4} H_0^{(1)}\left(k|\mathbf{r} - \mathbf{r}'|\right) \tag{23}$$

with $H_0^{(1)}(\cdot)$, which is the Hankel function of the first kind and zero order. Substituting (22) into (21) and using the jump relation of the single layer potential, one gets the boundary integral equation for unknown φ as

$$(S\varphi)(\mathbf{r}) + \frac{\eta}{ik}(K\varphi)(\mathbf{r}) - \frac{\eta}{2ik}(\varphi)(\mathbf{r}) = -u^i(\mathbf{r}) - \frac{\eta}{ik}\frac{\partial u^i}{\partial n}(\mathbf{r}), \tag{24}$$

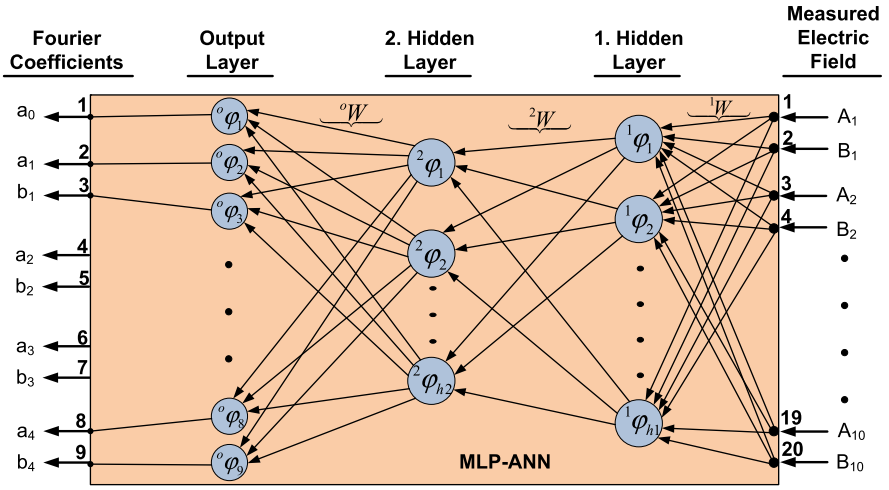


Fig. 7 Constituting inverse coarse model for scattering problem using MLP-ANN

$\mathbf{r} \in \partial D$, where S and K are the single and double layer operators given by

$$(S\varphi)(\mathbf{r}) = \int_{\partial D} \varphi(\mathbf{r}')\Phi(\mathbf{r}, \mathbf{r}')\, ds(\mathbf{r}'), \tag{25}$$

$$(K\varphi)(\mathbf{r}) = \frac{\partial}{\partial n(\mathbf{r})} \int_{\partial D} \varphi(\mathbf{r}')\Phi(\mathbf{r}, \mathbf{r}')\, ds(\mathbf{r}'). \tag{26}$$

The boundary integral equation in (24) can be solved numerically by using a Nyström method that has an exponential convergence property [11, 12].

5.2 *Inverse Scattering Problem with High Dimension*

The inverse scattering problem involves obtaining the shape of a conductor cylinder using electromagnetic field measurements. This problem will be calculated using the ANN, ASM, and SM-ID methods, to give a detailed comparison between them. The conductance cylinder is illuminated by a TMz wave with frequency 33 MHz and angle of incidence $\phi_i = 0$. The scattered field data is measured at 10 points on a measurement circle with radius 100λ as indicated in Fig. 5. The shape of the conductor cylinder is represented by one real and four complex Fourier coefficients, as indicated in Fig. 6. The inverse coarse model used has 20 inputs for complex measurement data and 9 outputs for Fourier coefficients.

The ANN is used not only as one method for solving the problem but also as a way of building the coarse model. The considered ANN structure as indicated in Fig. 7 is a feedforward multi-layer perceptron (MLP) with two hidden layers having "20" and "30" neurons, respectively. The ANN is trained with different numbers

of data, such as 25, 50, and 100. Therefore, inverse coarse models with different accuracies are generated to investigate the efficiency of the SM-ID technique depending on the coarse model. Normally ASM utilizes parameter extraction to find coarse model parameters, but to convey equality in comparing the results obtained by SM-ID and ASM, the inverse coarse model is also used in ASM.

In this work, the high inverse scattering problem is solved for two different geometries. The shapes of the conductor cylinder obtained by ANN, ASM, and SM-ID are given for three inverse coarse models. Different accuracies of the inverse coarse model affect the convergence and accuracy of the desired fine model parameters. To compare accuracy, it is necessary to use error measures. The mean and maximum error are determined by (27) and (28):

$$Mean\ Error = \frac{1}{360} \times \sum_{i=1}^{360} \frac{|X_{\text{original},i} - X_{\text{method},i}|}{X_{\text{original},i}}, \tag{27}$$

$$Max\ Error = \max_i \left\{ \frac{|X_{\text{original},i} - X_{\text{method},i}|}{X_{\text{original},i}} \right\}. \tag{28}$$

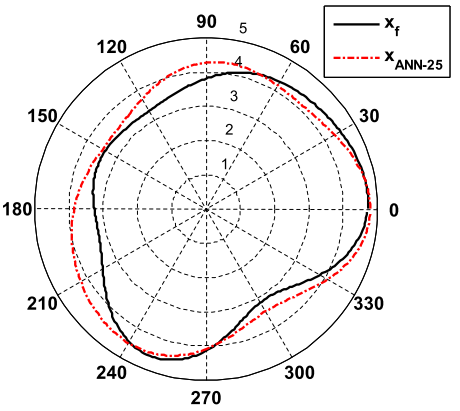
The stopping criterion is satisfied if the maximum error between the target fine model response and the fine model response corresponding to the obtained fine model parameter is less than 0.003. This corresponds to a maximum error of less than 0.003. However, the maximum and mean error calculations are different from this stopping criterion based on maximum error as defined above. The maximum and mean error calculations depend on the difference between the desired fine model parameters and the obtained fine model parameters; hence, the mean error can be greater than the stopping criterion.

The results of the shape reconstruction using optimization techniques for two geometries are given in two parts. The first part includes shape reconstructions for geometry 1 obtained from ANN, ASM, and SM-ID. The ANN results for different inverse coarse models are shown in Fig. 8. The ASM and SM-ID results are given in Fig. 9 and Fig. 10, respectively.

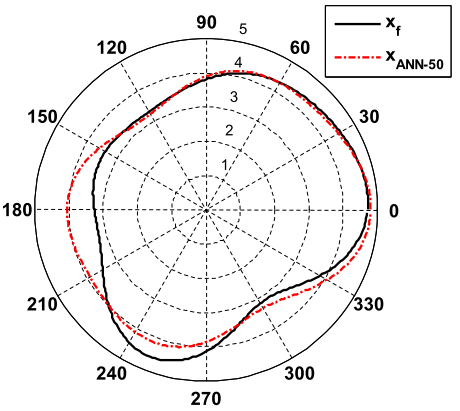
The second part includes shape reconstructions for geometry 2. The ANN results for different inverse coarse models are shown in Fig. 11, and the ASM and SM-ID results are shown in Fig. 12 and Fig. 13, respectively. The result obtained from the 13th iteration is shown in Fig. 12(a) instead of the original shape because of the divergence of *ASM*, 25. The original solution is given in Table 1.

The convergence of ASM and SM-ID obtained from inverse coarse models trained with three sets including 25, 50, and 100 data, respectively, for geometry 1 is given in Fig. 14. When the accuracy of the inverse coarse model reduces due to less data usage for training, the required iteration step increases. In contrast, SM-ID is not badly affected by a less accurate inverse coarse model, since SM-ID exploits inverse difference mapping based on more knowledge obtained from the fine model execution.

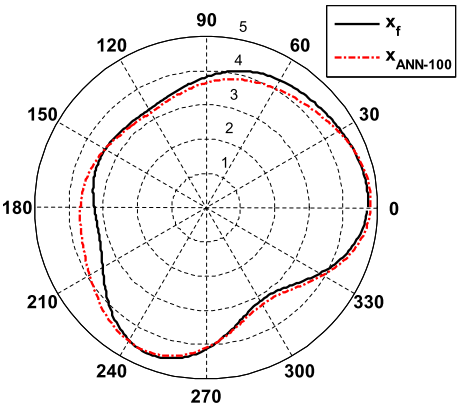
The convergence of ASM and SM-ID obtained from inverse coarse models trained with three sets including 25, 50, and 100 data, respectively, for geometry



(a)

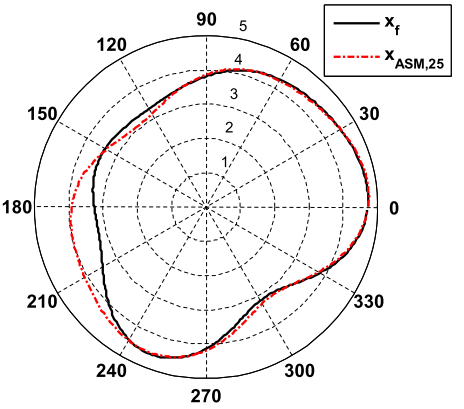


(b)

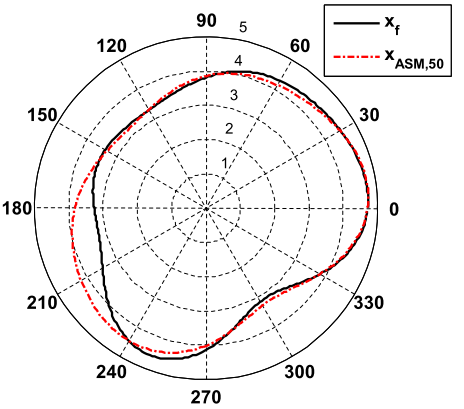


(c)

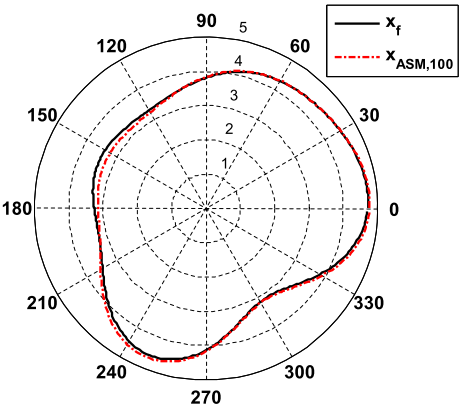
Fig. 8 The shape reconstruction obtained from coarse models trained with three sets including 25, 50, and 100 data, respectively: (**a**) *ANN*-25, (**b**) *ANN*-50, and (**c**) *ANN*-100 for geometry 1



(a)

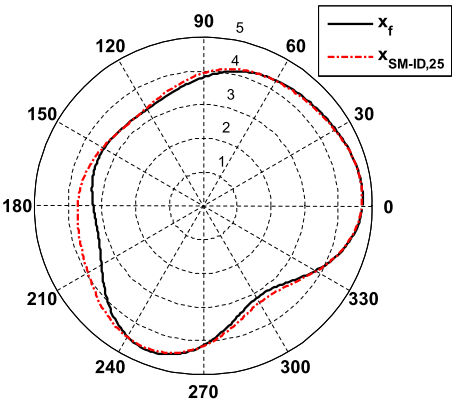


(b)

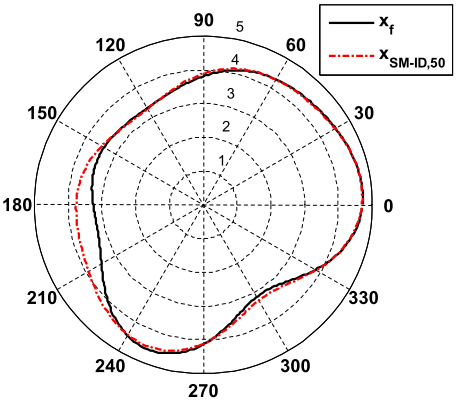


(c)

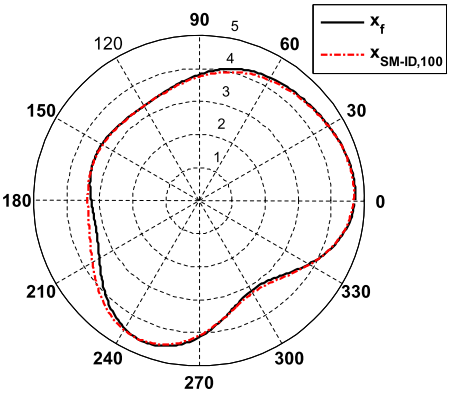
Fig. 9 The shape reconstruction obtained from ASM model for different inverse coarse models trained with three sets including 25, 50, and 100 data, respectively: (**a**) *ASM*, 25, (**b**) *ASM*, 50, and (**c**) *ASM*, 100 for geometry 1



(a)

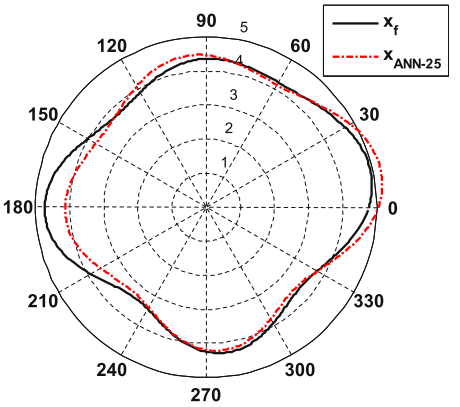


(b)

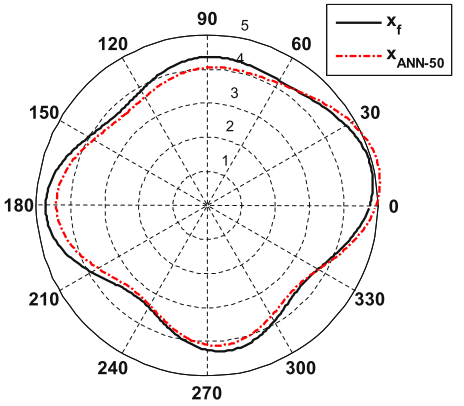


(c)

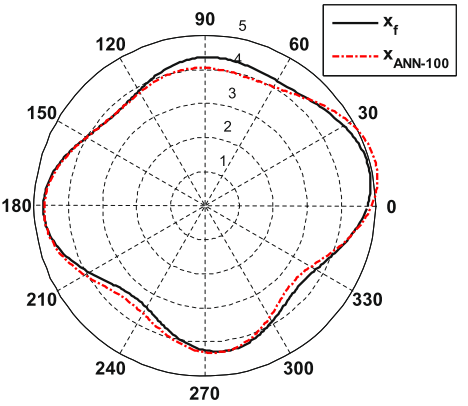
Fig. 10 The shape reconstruction obtained from SM-ID model for different inverse coarse models trained with three sets including 25, 50, and 100 data, respectively: (**a**) *SM-ID*, 25, (**b**) *SM-ID*, 50, and (**c**) *SM-ID*, 100 for geometry 1



(a)

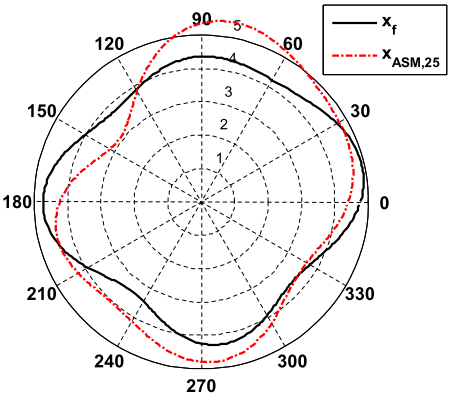


(b)

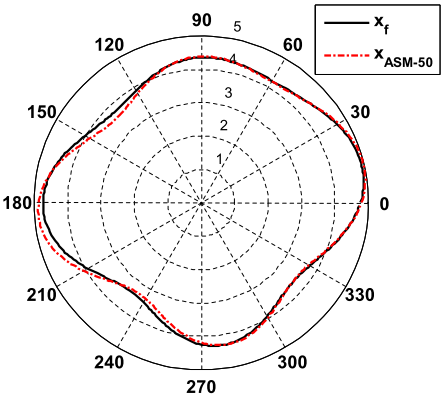


(c)

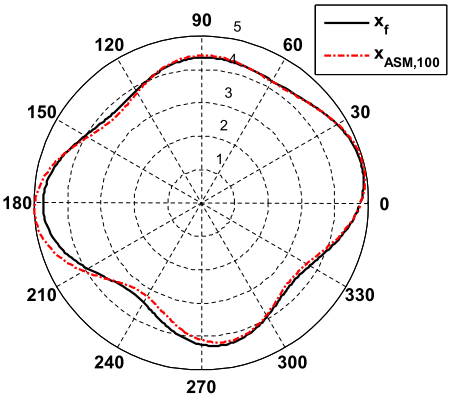
Fig. 11 The shape reconstruction obtained from coarse models trained with three sets including 25, 50, and 100 data, respectively: (**a**) *ANN*-25, (**b**) *ANN*-50, and (**c**) *ANN*-100 for geometry 2



(a)

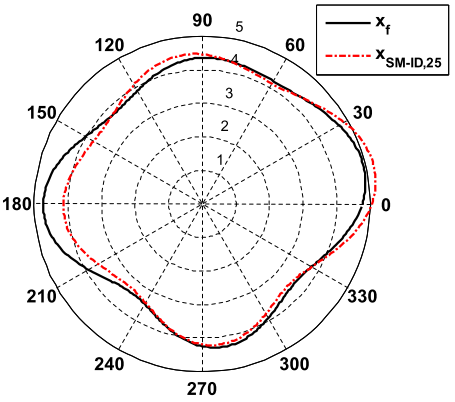


(b)

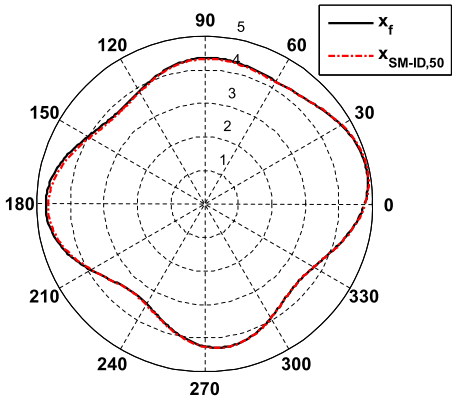


(c)

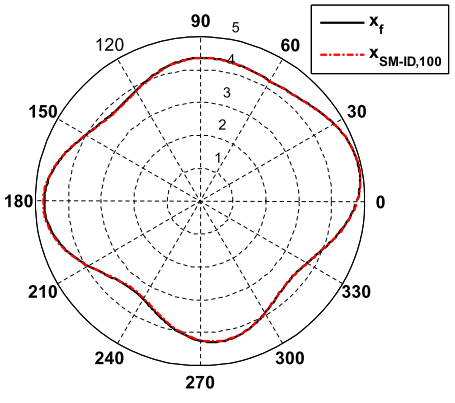
Fig. 12 The shape reconstruction obtained from ASM model for different inverse coarse models trained with three sets including 25, 50, and 100 data, respectively: (**a**) *ASM*, 25, (**b**) *ASM*, 50, and (**c**) *ASM*, 100 for geometry 2



(a)



(b)

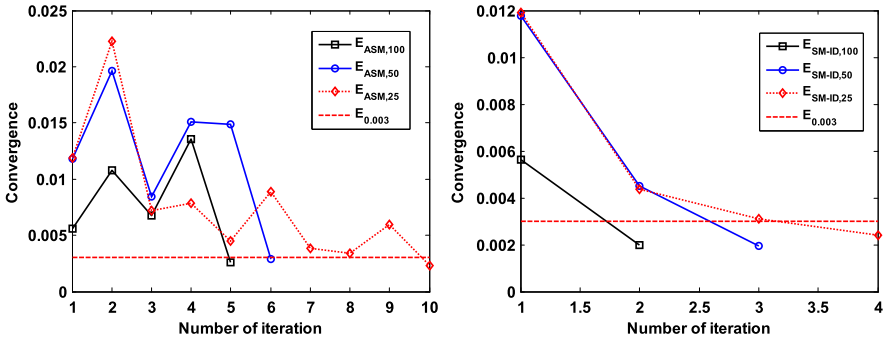


(c)

Fig. 13 The shape reconstruction obtained from SM-ID model for different inverse coarse models trained with three sets including 25, 50, and 100 data, respectively: (**a**) *SM-ID*, 25, (**b**) *SM-ID*, 50, and (**c**) *SM-ID*, 100 for geometry 2

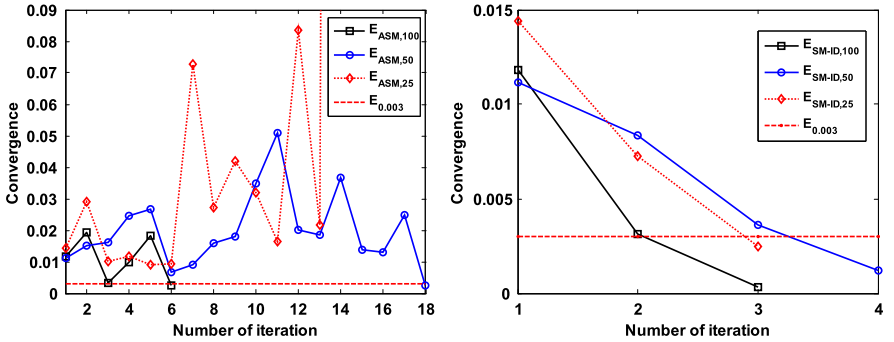
Table 1 Overall results for geometry 1 and geometry 2 obtained from ANN, ASM, and SM-ID

Method	Geometry 1			Geometry 2		
	Iteration number	Mean error	Max error	Iteration number	Mean error	Max error
ANN-25	–	0.07899	0.25313	–	0.04413	0.12717
ANN-50	–	0.06081	0.25466	–	0.04202	0.07488
ANN-100	–	0.04331	0.15857	–	0.03175	0.07912
ASM, 25	10	0.04562	0.24170	18	321.741	1182.46
ASM, 50	6	0.04729	0.24891	18	0.01681	0.05040
ASM, 100	5	0.01608	0.05203	6	0.02290	0.06720
SM-ID, 25	4	0.03621	0.17425	3	0.03258	0.12518
SM-ID, 50	3	0.03384	0.17799	4	0.00591	0.01808
SM-ID, 100	2	0.01676	0.07092	3	0.00309	0.01041



(a) (b)

Fig. 14 For different inverse coarse models with different accuracies, the convergence of (**a**) *ASM* and (**b**) *SM-ID* for geometry 1



(a) (b)

Fig. 15 For different inverse coarse models with different accuracies, the convergence of (**a**) *ASM* and (**b**) *SM-ID* for geometry 2

2 is given in Fig. 15. The required iteration steps of ASM increase with the lower accuracy of the inverse coarse model. With only a few data sets used in the inverse coarse model, like the one trained with 25 data, ASM diverges as shown in Fig. 15(a). In contrast, SM-ID is not badly affected by a less accurate inverse coarse model, since SM-ID exploits inverse difference mapping based on more knowledge obtained from the fine model execution. Because in SM-ID a new iteration point is determined not only from optimum coarse model parameters but also by using the target fine model response, SM-ID is robust against more discrepancy between the coarse model and the fine model.

6 Conclusion

The difference mapping concept is considered from the convergence point of view in this work. The power of difference mapping based on the PKI-D technique is expressed with mathematical representations. To demonstrate the efficiency of difference mapping, SM-ID is compared with ASM and ANN in solving a high inverse scattering problem. High dimensions increase the complexity of the problem, so the results from two different geometries are used to demonstrate the difference mapping advantage explicitly. Different inverse coarse models with different accuracies are generated to show the efficiency of SM-ID compared with ASM. The robustness of SM-ID against less accurate coarse models is observed from the figures and table. Moreover, it is observed that SM-ID requires fewer iteration steps than ASM to solve the high inverse scattering problem.

References

1. Akduman, I., Kress, R.: Direct and inverse scattering problems for inhomogeneous impedance cylinders of arbitrary shape. Radio Sci. **38**, 1055–1064 (2003)
2. Bakr, M.H., Bandler, J.W., Georgieva, N.: An aggressive approach to parameter extraction. IEEE Trans. Microw. Theory Tech. **47**(12), 2428–2439 (1999)
3. Bakr, M.H., Bandler, J.W., Madsen, K., Rayas-Sanchez, J.E., Sondergaard, J.: Space-mapping optimization of microwave circuit exploiting surrogate models. IEEE Trans. Microw. Theory Tech. **48**(2), 2297–2306 (2000)
4. Bakr, M.H., Bandler, J.W., Madsen, K., Sondergaard, J.: An introduction to the space mapping technique. Optim. Eng. **2**, 369–384 (2001)
5. Bandler, J.W., Biernacki, R.M., Chen, S.H., Grobelny, P.A., Hemmers, R.H.: Space mapping technique for electromagnetic optimization. IEEE Trans. Microw. Theory Tech. **48**(12), 2536–2544 (1994)
6. Bandler, J.W., Biernacki, R.M., Chen, S.H., Hemmers, R.H., Madsen, K.: Electromagnetic optimization exploiting aggressive space mapping. IEEE Trans. Microw. Theory Tech. **43**(12), 2874–2882 (1995)
7. Bandler, J.W., Cheng, Q.S., Dakroury, S.A., Mohamed, A.S., Bakr, M.H., Madsen, K., Sondergaard, J.: Space mapping: the state of the art. IEEE Trans. Microw. Theory Tech. **52**(1), 337–361 (2004)

8. Benedetti, M., Lesselier, D., Lambert, M., Massa, A.: A multi-resolution technique based on shape optimization for the reconstruction of homogeneous dielectric objects. Inverse Probl. **25**, 1–26 (2009)
9. Booker, A.J., Dennis, J.E. Jr., Frank, P.D., Serafini, D.B., Torczon, V., Trosset, M.W.: A rigorous framework for optimization of expensive function by surrogates. Struct. Optim. **17**, 1–13 (1999)
10. Christodoulou, C., Georgiopoulos, M.: Applications of Neural Networks in Electromagnetics. Artech House, Boston (2001)
11. Colton, D., Kress, R.: Inverse Acoustic and Electromagnetic Scattering Theory. Wiley-Interscience, New York (1983)
12. Colton, D., Kress, R.: Inverse Acoustic and Electromagnetic Scattering Theory, 2nd edn. Springer, Berlin (1999)
13. Koziel, S., Bandler, J.W.: Space mapping algorithm with improved convergence properties for microwave design optimization. Int. J. RF Microw. Comput.-Aided Eng. **20**(2), 230–240 (2010)
14. Rekanos, I.T.: Shape reconstruction of a perfectly conducting scatterer using differential evolution and particle swarm optimization. IEEE Trans. Geosci. Remote Sens. **46**, 1967–1974 (2008)
15. Simsek, M., Sengor, N.S.: A new modeling method based on difference between fine and coarse models using space mapping. In: SMSMEO-06 Conference Proceedings, Technical University of Denmark, Copenhagen, Denmark, 9–11 Nov. 2006, pp. 64–65 (2006)
16. Simsek, M., Sengor, N.S.: A knowledge-based neuromodeling using space mapping technique: compound space mapping-based neuromodeling. Int. J. Numer. Model. **21**(1–2), 133–149 (2008)
17. Simsek, M., Sengor, N.S.: An efficient inverse ANN modeling approach using prior knowledge input with difference method. In: The European Conference on Circuit Theory and Design, Antalya, Turkey, August 23–27 (2009)
18. Simsek, M., Sengor, N.S.: Solving inverse problems by space mapping with inverse difference method. In: Roos, J., Costa, L.R.J. (eds.) Scientific Computing in Electrical Engineering SCEE 2008. Mathematics in Industry, vol. 14, pp. 453–460. Springer, Heidelberg (2010)
19. Simsek, M., Tezel, N.S.: The reconstruction of shape and impedance exploiting space mapping with inverse difference method. IEEE Trans. Antennas Propag. **60**(4), 1868–1877 (2012)
20. Simsek, M., Zhang, Q.J., Kabir, H., Cao, Y., Sengor, N.S.: The recent developments in microwave design. Int. J. Math. Model. Numer. Optim. **2**(2), 213–228 (2011)
21. Sondergard, J.: Optimization using surrogate models by the space mapping technique. Ph.D. thesis, Informatics and Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, Jan. (2003)
22. Zhang, Q.J., Gupta, K.C.: Neural Networks for RF and Microwave Design. Artech House, Boston (2000)

Bayesian Support Vector Regression Modeling of Microwave Structures for Design Applications

J. Pieter Jacobs, Slawomir Koziel, and Leifur Leifsson

Abstract Fast and accurate models are indispensable in contemporary microwave engineering. Kernel-based machine learning methods applied to the modeling of microwave structures have recently attracted substantial attention; these include support vector regression and Gaussian process regression. Among them, Bayesian support vector regression (BSVR) with automatic relevance determination (ARD) proved to perform particularly well when modeling input characteristics of microwave devices. In this chapter, we apply BSVR to the modeling of microwave antennas and filters. Moreover, we discuss a more efficient version of BSVR-based modeling exploiting variable-fidelity electromagnetic (EM) simulations, where coarse-discretization EM simulation data is used to find a reduced number of fine-discretization training points for establishing a high-fidelity BSVR model of the device of interest. We apply the BSVR models to design optimization. In particular, embedding the BSVR model obtained from coarse-discretization EM data into a surrogate-based optimization framework exploiting space mapping allows us to yield an optimized design at a low computational cost corresponding to a few evaluations of the high-fidelity EM model of the considered device. The presented techniques are illustrated using several examples of antennas and microstrip filters.

Keywords Bayesian support vector regression · Computer-aided design (CAD) · Electromagnetic (EM) simulation · Microwave engineering · Space mapping · Surrogate-based optimization · Surrogate modeling

J.P. Jacobs (✉)
Centre for Electromagnetism, Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria, 0002, South Africa
e-mail: jpjacobs@up.ac.za

S. Koziel · L. Leifsson
Engineering Optimization & Modeling Center, School of Science and Engineering, Reykjavik University, Menntavegur 1, 101 Reykjavik, Iceland

S. Koziel
e-mail: koziel@ru.is

L. Leifsson
e-mail: leifurth@ru.is

S. Koziel, L. Leifsson (eds.), *Surrogate-Based Modeling and Optimization*,
DOI 10.1007/978-1-4614-7551-4_6,

1 Introduction

Full-wave electromagnetic (EM) simulations based on the method of moments and/or finite elements play a ubiquitous part in microwave engineering, as they permit highly accurate evaluation of microwave structures such as planar antennas and filters. Such simulations, however, are costly in computational terms, and their use for tasks requiring numerous analyses (e.g., statistical analysis and parametric design optimization) might become infeasible under certain conditions (for instance, a genetic algorithm optimization might necessitate thousands of full-wave analyses of candidate geometries of the structure to be optimized). Hence, surrogate models are used instead. Trained on a training set consisting of a limited number of input-output pairs (such as adjustable antenna geometry parameters and frequency as input, and the magnitude of the input reflection coefficient $|S_{11}|$ obtained from full-wave simulations as output), these models, by virtue of their ability to generalize over the input space, make it possible to quickly obtain the desired performance characteristics for inputs not previously presented to the model.

The kernel-based machine learning method most widely used for microwave modeling tasks has been support vector regression (SVR) utilizing an isotropic Gaussian kernel [1]. It has recently been shown [2] that Bayesian support vector regression (BSVR) [3] using a Gaussian kernel with automatic relevance determination (ARD) significantly outperforms the above standard SVR with an isotropic kernel in modeling $|S_{11}|$ versus the frequency of CPW-fed slot antennas with multiple tunable geometry variables. BSVR is in essence a version of Gaussian process regression (GPR) [4]; the Bayesian framework enables efficient training of the multiple hyperparameters of the ARD kernel by minimizing the negative log probability of the data given the hyperparameters. Such training of multiple hyperparameters is intractable under standard SVR, which employs a grid-search/cross-validation approach towards this end. In addition to its advantageous Bayesian-based features, BSVR also exhibits certain desirable properties of standard SVR, such as quadratic programming and sparseness of solutions, i.e., solutions that are fully characterized by the set of SVs, which is a subset of the training set.

In this chapter, we explore BSVR within both global and local modeling contexts. Global, or "library-type," surrogate models aim at giving accurate predictions over the entire input space, and can be used for a variety of applications (e.g., optimization and statistical analysis). In contrast, local/trust region models only apply to a subregion of input space specified by the optimization algorithm within which the model usually is embedded.

Similar to many other global modeling methods, a drawback of BSVR is the high starting cost of gathering the fine-discretization full-wave simulation data necessary to train the model so that it has high predictive accuracy. We address this problem by exploiting the sparseness property of BSVR to reduce the amount of expensive high-fidelity data required for training (see Sects. 3–5). Earlier methods aimed at optimal data selection for microwave modeling problems include various adaptive sampling techniques that aim, within optimization contexts, to reduce the number of samples necessary to ensure the desired modeling accuracy. This is done by iterative

identification of the model and the addition of new training samples based on the actual model error at selected locations (e.g., [5]) or expected error values (statistical infill criteria, e.g., [6]); [5, 6] were local/trust region models.

Our approach entails first training an auxiliary BSVR model using fast, inexpensive coarse-discretization data selected by means of traditional experimental design procedures, and then taking the support vectors of this model simulated at a high mesh density as training data for the actual (high-fidelity) BSVR model. (A similar approach was adopted in [7], but only standard SVR with an isotropic kernel was used to model comparatively uncomplicated underlying functions.) The role of the auxiliary model can be viewed as locating regions of the design space where more samples are needed compared to other regions—for example, because the response is more variable with respect to the design and/or frequency variables. Our modeling approach is demonstrated using both planar antenna and microstrip filter examples (see Sects. 4 and 5, respectively). We also evaluate the accuracy of our reduced-data BSVR surrogates by using them within a space mapping (SM) optimization framework.

As to local BSVR modeling, we consider surrogates for variable-fidelity EM-driven optimization (see Sect. 6). In this approach, the optimization is carried out using SM, whereas the underlying coarse model is created by approximating coarse-discretization EM simulation data using BSVR. The high-fidelity EM simulation is only launched to verify the design produced by the space-mapped BSVR coarse model and obtain the data for its further correction. This allows us to significantly reduce the computational cost of the design optimization process as illustrated using two antenna examples.

The above sections are preceded by a short overview in Sect. 2 of the BSVR framework, and followed by some summary remarks (see Sect. 7).

2 Modeling Using Bayesian Support Vector Regression

In this section, we briefly give an overview of the formulation of Bayesian support vector regression (BSVR).

Consider a training data set of n observations, $\mathcal{D} = \{(\boldsymbol{u}_i, y_i) \mid i = 1, \ldots, n\}$. The BSVR formulation, which is explained at length in [3], follows the standard Bayesian regression framework for GPR in which training targets y_i corresponding to input vectors $\boldsymbol{u}_i$ are expressed as $y_i = f(\boldsymbol{u}_i) + \delta_i$, where δ_i are independent, identically distributed noise variables; and the underlying function f is a random field. If $\mathbf{f} = [f(\boldsymbol{u}_1)\ f(\boldsymbol{u}_2)\ \ldots\ f(\boldsymbol{u}_n)]$, then Bayes's theorem gives the posterior probability of $\mathbf{f}$ given the training data $\mathcal{D}$ as

$$p(\mathbf{f}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{f})\,p(\mathbf{f})}{p(\mathcal{D})}, \tag{1}$$

with $p(\mathbf{f})$ the prior probability of $\mathbf{f}$, $p(\mathcal{D}|\mathbf{f})$ the likelihood, and $p(\mathcal{D})$ the evidence. The likelihood is given by

$$p(\mathcal{D}|\mathbf{f}) = \prod_{i=1}^{n} p(\delta_i), \tag{2}$$

where $p(\delta_i) \propto \exp(-\zeta L(\delta_i))$with $L(\delta_i)$ the loss function, and ζ a constant. In standard GPR [4] the loss function is quadratic; the crucial point in the BSVR formulation is that a new loss function, the soft insensitive loss function, is used that combines advantageous properties of both the ε-insensitive loss function (sparseness of solutions) of standard SVR [8], and Huber's loss function (differentiability). It is defined as [3]:

$$L_{\varepsilon,\beta}(\delta) = \begin{cases} -\delta - \varepsilon; & \delta \in (-\infty, -(1+\beta)\varepsilon), \\ \frac{(\delta+(1-\beta)\varepsilon)^2}{4\beta\varepsilon}; & \delta \in [-(1+\beta)\varepsilon, -(1-\beta)\varepsilon], \\ 0; & \delta \in (-(1-\beta)\varepsilon, (1-\beta)\varepsilon), \\ \frac{(\delta-(1-\beta)\varepsilon)^2}{4\beta\varepsilon}; & \delta \in [(1-\beta)\varepsilon, (1+\beta)\varepsilon], \\ \delta - \varepsilon; & \delta \in ((1+\beta)\varepsilon, +\infty), \end{cases} \tag{3}$$

where $0 < \beta \leq 1$, and $\varepsilon > 0$.

Solving for the maximum a posteriori (MAP) estimate of the function values entails solving the primal problem [3, Eqs. (19)–(21)], with the corresponding dual problem given by

$$\begin{aligned} \min_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} \quad & \frac{1}{2}\left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\right)^\top \Sigma \left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\right) - \sum_{i=1}^{n} y_i \left(\alpha_i - \alpha_i^*\right) \\ & + \sum_{i=1}^{n} \left(\alpha_i + \alpha_i^*\right)(1-\beta)\varepsilon + \frac{\beta\varepsilon}{C} \sum_{i=1}^{n} y_i \left(\alpha_i^2 + \alpha_i^{*2}\right) \end{aligned} \tag{4}$$

subject to $0 \leq \alpha_i, \alpha_i^* \leq C$, $i = 1, \ldots, n$. In the above, Σ is an $n \times n$ matrix with $\Sigma_{ij} = k(\boldsymbol{u}_i, \boldsymbol{u}_j)$ and $k(\cdot)$ is the kernel function. In particular, the Gaussian kernel with ARD (used throughout in this work) is given by

$$k(\boldsymbol{u}_i, \boldsymbol{u}_j) = \sigma_f^2 \exp\left(-\frac{1}{2}\sum_{k=1}^{D} \frac{(u_{ik} - u_{jk})^2}{\tau_k^2}\right) + \kappa, \tag{5}$$

where u_{ik} and u_{jk} are the kth elements of the ith and jth training input vectors. The hyperparameter vector $\boldsymbol{\theta}$, which includes σ_f^2, τ_k, κ, C, and ε, can be determined by minimizing the negative log probability of the data given the hyperparameters [3],

$$\begin{aligned} -\ln p(\mathcal{D}|\boldsymbol{\theta}) = & \frac{1}{2}\left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\right)^\top \Sigma \left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\right) \\ & + C \sum_{i=1}^{n} L_{\varepsilon,\beta}\left(y_i - f_{\mathrm{MP}}(x_i)\right) + \frac{1}{2} \ln \left| \boldsymbol{I} + \frac{C}{2\beta\varepsilon} \Sigma_M \right| + n \ln Z_S, \end{aligned} \tag{6}$$

with Σ_M an $m \times m$ submatrix of Σ corresponding to the off-bound support vectors, $\boldsymbol{I}$ the $m \times m$ identity matrix, $\mathbf{f}_{\mathrm{MP}} = \Sigma(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)$, and Z_s defined as [3, Eq. (15)]. The length scale τ_k associated with the kth input dimension can be considered the distance that has to be traveled along that dimension before the output changes significantly [4]. The regression estimate at a test input $\boldsymbol{u}^*$ can be expressed as

$$f(\boldsymbol{u}^*) = \sum_{i=1}^{n} k(\boldsymbol{u}_i, \boldsymbol{u}^*)(\alpha_i - \alpha_i^*). \tag{7}$$

Training points corresponding to $|\alpha_i - \alpha_i^*| > 0$ are the support vectors (SVs); of these, points corresponding to $0 < |\alpha_i - \alpha_i^*| < C$ are termed off-bound SVs. Usually, the lower the parameter β in the loss function, the smaller the number of SVs [3]; β determines the density function of the additive noise associated with training targets.

3 BSVR Modeling with Reduced Data Sets

In this section, we discuss a method of exploiting EM simulations of variable fidelity in order to reduce the computational cost of creating the BSVR model. The model response of interest typically is $|S_{11}|$ or $|S_{21}|$ over a specified frequency range for a particular antenna or filter geometry. A model input (column) vector $\boldsymbol{u}$ consists of the set of adjustable geometry parameters $\boldsymbol{u}$ and a frequency value f; thus we have $\boldsymbol{u} = [\boldsymbol{x}^T f]^T$. The (scalar) response of a model at a specific frequency, for example, the model response $\boldsymbol{R}_f$, which is the fine-discretization full-wave simulated S-parameters, is denoted as $\boldsymbol{R}_f(\boldsymbol{u})$, or $\boldsymbol{R}_f(\boldsymbol{x}, f)$. Suppose now that a BSVR surrogate $\boldsymbol{R}_s$ of the CPU-intensive high-fidelity model $\boldsymbol{R}_f$ has to be constructed. As noted earlier, the computational cost of gathering sufficient data to train $\boldsymbol{R}_s$ typically is high. To address this, first, an auxiliary BSVR model $\boldsymbol{R}_{s.\mathrm{aux}}$ of the antenna (or filter) is set up with training data obtained from coarse-discretization full-wave simulations (these simulations are referred to as the low-fidelity full-wave model $\boldsymbol{R}_c$). The training set for $\boldsymbol{R}_{s.\mathrm{aux}}$ consists of n input vectors $\boldsymbol{x}_i$, $i = 1, \ldots, n$, and associated targets $y_i = \boldsymbol{R}_c(\boldsymbol{u}_i)$, where $\boldsymbol{u}_i$ contains geometry parameters and a frequency value as noted above, and y_i is the corresponding simulated $|S_{11}|$ (or $|S_{21}|$) value. The SVs obtained from $\boldsymbol{R}_{s.\mathrm{aux}}$ are then simulated at the (high) mesh density of $\boldsymbol{R}_f$, providing the reduced fine-discretization training set for $\boldsymbol{R}_s$.

As experience has shown that the coarsely simulated targets $\boldsymbol{R}_c$ and finely simulated targets $\boldsymbol{R}_f$ of microwave structures such as antennas or filters are usually reasonably well correlated (in Sects. 4 and 5 we give coarse and fine meshing densities for specific examples to indicate by how much they can differ), we assume that the regions of the input space which support the crucial variations in the coarse response surface will also support the crucial variations in the fine response surface. Hence, the SVs of the coarse model should largely capture the crucial variations in the fine data as well, and along with target values obtained through fine-discretization simulations should make an adequate reduced-data training set for a high-fidelity BSVR model, i.e., $\boldsymbol{R}_s$.

4 Modeling and Optimization of Antennas Using BSVR

In this section, we present examples illustrating how global BSVR models for the reflection coefficients of planar slot antennas can be set up based on reduced finely discretized data sets. We then use these models for design optimization. We consider three examples of antennas with highly nonlinear $|S_{11}|$ responses as a function of tunable geometry parameters and frequency: a narrowband coplanar waveguide (CPW)-fed slot dipole antenna, an ultra-wideband (UWB) CPW-fed T-shaped slot antenna, and a broadband probe-fed microstrip patch with two U-shaped parasitic elements. We furthermore evaluate the accuracy of our reduced-data BSVR surrogates by using them within a space mapping (SM) optimization framework [9–14].

4.1 Slot Dipole Antenna (Antenna 1)

Figure 1 shows the geometry of a CPW-fed slot dipole antenna on a single-layer dielectric substrate. The design variables were $\boldsymbol{x} = [W \;\; L]^T$ mm, and the input space was specified as $5 \leq W \leq 10$ mm and $28 \leq L \leq 50$ mm. Other dimensions/parameters were $w_0 = 4.0$ mm, $s = 0.5$ mm, $h = 1.6$ mm, and $\varepsilon_r = 4.4$. We were concerned with $|S_{11}|$ over the frequency band 2.0–2.7 GHz (visual inspection revealed that $|S_{11}|$-versus-frequency responses over this band varied substantially throughout the above geometry input space). Using CST Microwave Studio [15] on a dual-core 2.33 GHz Intel CPU with 2 GB RAM, we considered a high-fidelity model $\boldsymbol{R}_f$ (∼130,000 mesh cells, simulation time 12 min) and a low-fidelity model $\boldsymbol{R}_c$ (∼5,000 mesh cells, simulation time 30 s).

For training input data, 99 geometries were selected at random from the input space using Latin hypercube sampling (LHS), with three frequencies per geometry uniformly randomly sampled from the above frequency range such that, in general, each geometry had a different set of frequencies. The total number of training points was $n = 99 \times 3 = 297$; training input vectors had the form $\{\boldsymbol{u}_i = [\boldsymbol{x}_i^T \;\; f_i]^T = [W_i \;\; L_i \;\; f_i]^T | i = 1, \ldots, n\}$, with W_i and L_i the design variables corresponding to the ith input vector, and f_i a frequency value within the range of interest. Test data consisted of 100 new geometries, also obtained via LHS, with 71 equally spaced frequencies per geometry. The training data were simulated at the $\boldsymbol{R}_c$ mesh density, and used to train the BSVR model $\boldsymbol{R}_{s,\text{aux}}$ for three different values of β at the low end of its possible range ($\beta \in \{0.05, 0.15, 0.25\}$; as noted earlier, usually the smaller the value of β, the smaller the number of SVs). Each $\boldsymbol{R}_{s,\text{aux}}$ was used to predict the test data (also simulated at the $\boldsymbol{R}_c$ mesh density). %RMSE (percentage root mean square error normalized to the target range) values were in the vicinity of 1.1 %; this high predictive accuracy confirmed that the training set was sufficiently large.

For each $\boldsymbol{R}_{s,\text{aux}}$ model, the SVs were identified and simulated at the $\boldsymbol{R}_f$ mesh density. BSVR models fitted to these fine-discretization data gave the desired surrogate models $\boldsymbol{R}_s$. For comparison, surrogate models $\boldsymbol{R}_{s,\text{full}}$ trained on the full fine-discretization training data set ($n = 297$) were also set up. Table 1 gives, for each of

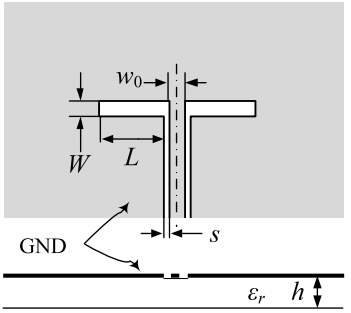


Fig. 1 Geometry of a CPW-fed slot dipole antenna (Antenna 1). The ground plane (GND) has infinite lateral extent

Table 1 Predictive errors of surrogate antenna models

Antenna	β	RMSE (%)			n_{SV}	n_{SV}/n (%)	Model cost[a]	
		$\boldsymbol{R}_{s,\text{aux}}$	$\boldsymbol{R}_s$	$\boldsymbol{R}_{s,\text{full}}$			$\boldsymbol{R}_s$[b]	$\boldsymbol{R}_{s,\text{full}}$
1 ($n = 297$)	0.05	1.14	1.77	1.68	168	56	168	297
	0.15	1.13	1.71	1.72	176	59	176	
	0.25	1.15	1.68	1.68	206	69	206	
2 ($n = 3{,}528$)	0.05	3.37	4.14	3.83	1,819	52	1,819	3528
	0.15	3.14	4.11	3.52	2,010	57	2,010	
	0.25	3.13	3.89	3.50	2,225	63	2,225	
3 ($n = 6{,}400$)	0.05	6.59	5.77	5.53	3,658	57	3,658	6,400
	0.15	6.54	5.87	5.41	3,937	62	3,937	
	0.25	6.59	5.71	5.27	3,916	61	3,916	

[a]Cost is in terms of number of $\boldsymbol{R}_f$ evaluations, i.e., fine-discretization full-wave simulations, required to generate the training data

[b]The actual cost of setting up $\boldsymbol{R}_s$ is slightly (by 1 to 4 percent depending on the particular antenna example) higher due to the coarse-discretization EM simulations used in the model construction process

the β values, the %RMSE values obtained with $\boldsymbol{R}_{s,\text{aux}}$ on the coarse-discretization test data, and $\boldsymbol{R}_s$ and $\boldsymbol{R}_{s,\text{full}}$ on the fine-discretization test data.

Also given in the table is n_{SV}, the number of SVs associated with $\boldsymbol{R}_{s,\text{aux}}$ (and therefore the number of training points for $\boldsymbol{R}_s$), and n_{SV}/n, the proportion of the full training data that were SVs of $\boldsymbol{R}_{s,\text{aux}}$. The %RMSE values obtained for $\boldsymbol{R}_s$ and $\boldsymbol{R}_{s,\text{full}}$ were either the same, or only marginally higher in the case of $\boldsymbol{R}_s$, indicating that reducing the number of training points from n to n_{SV} by using the SVs of $\boldsymbol{R}_{s,\text{aux}}$ as training points for $\boldsymbol{R}_s$ incurred insignificant accuracy loss. In all cases, the reduction in training data was considerable: for example, for $\beta = 0.15$ the number of SVs was 176, which is 59 % of the original training data set. For ready comparison, Table 1 also explicitly lists the computational cost of generating the training data

for the models, expressed in terms of the number of fine-discretization simulations $\boldsymbol{R}_f$ (for each model it simply equals the number of training points). In terms of total CPU time (which was proportional to the costs in the table), these numbers translate to about 12 h for $\boldsymbol{R}_s$ ($\beta = 0.15$), and 20 h for $\boldsymbol{R}_{s,\text{full}}$.

4.2 UWB T-Shaped Slot Antenna (Antenna 2)

Figure 2 shows the antenna layout [16]. The design variables were $\boldsymbol{x} = [a_x \ a_y \ a \ b]^T$ mm, with design space $35 \leq a_x \leq 45$ mm, $20 \leq a_y \leq 35$ mm, $2 \leq a \leq 12$ mm, and $10 \leq b \leq 30$ mm ($w_0 = 4.0$ mm, $s_0 = 0.3$ mm, $s_1 = 1.7$ mm; the single-layer substrate had height $h = 0.813$ mm and dielectric constant $\varepsilon_r = 3.38$). The frequency band of interest was 2–8 GHz (as before, visual inspection confirmed that $|S_{11}|$-versus-frequency responses varied substantially throughout the geometry input space). Using CST Microwave Studio [15], we considered a high-fidelity model $\boldsymbol{R}_f$ (∼2,962,000 mesh cells, simulation time 21 min) and a low-fidelity model $\boldsymbol{R}_c$ (∼44,500 mesh cells, simulation time 20 s).

The training data consisted of 294 geometries obtained by LHS, with 12 frequencies per geometry, randomly selected as before ($n = 3{,}528$). The test data comprised 49 new LHS geometries, with 121 equally spaced frequencies per geometry (as before, the value of n was determined by the performance of $\boldsymbol{R}_{s,\text{aux}}$ on the test data simulated at the coarse mesh density).

The surrogate models $\boldsymbol{R}_{s,\text{aux}}$, $\boldsymbol{R}_s$, and $\boldsymbol{R}_{s,\text{full}}$ were set up similarly to those for Antenna 1. Table 1 gives, for three values of β, the %RMSEs obtained with $\boldsymbol{R}_{s,\text{aux}}$ on the coarse test data and with $\boldsymbol{R}_{s,}$ and $\boldsymbol{R}_{s,\text{full}}$ on the fine test data; as well as counts of SVs.

In general, %RMSE values of $\boldsymbol{R}_s$ were only somewhat higher than those of $\boldsymbol{R}_{s,\text{full}}$, suggesting as before that reducing the number of training points from n to n_{SV} by using the SVs of $\boldsymbol{R}_{s,\text{aux}}$ as fine-discretization training points for $\boldsymbol{R}_s$ has little effect on prediction accuracy. The CPU time required to generate fine-discretization training data for $\boldsymbol{R}_s$ in the case $\beta = 0.05$ (i.e., the model used in the optimization below) was approximately 56 h; the CPU time for $\boldsymbol{R}_{s,\text{full}}$ was 103 h.

4.3 Microstrip Antenna with Parasitic Elements (Antenna 3)

Figure 3 shows the antenna geometry [17]. The design variables were $\boldsymbol{x} = [a \ b \ c \ d \ e]^T$ mm, with design space $14 \leq a \leq 22$ mm, $0.4 \leq b \leq 2$ mm, $0.4 \leq c \leq 2$ mm, $0.4 \leq d \leq 2$ mm, and $0.4 \leq e \leq 2$ mm. The main patch had dimensions $a_0 = 5.8$ mm and $b_0 = 13.1$ mm. The lateral dimensions of the dielectric material and the metal ground were $l_x = 20$ mm and $l_y = 25$ mm. The dielectric substrate height, h, was 0.4 mm, and its relative permittivity, ε_r, was 4.3. The feed pin offset from the main patch center, l_0, was 5.05 mm, and the pin was 0.5 mm in diameter. The frequency band of interest was 4–7 GHz.

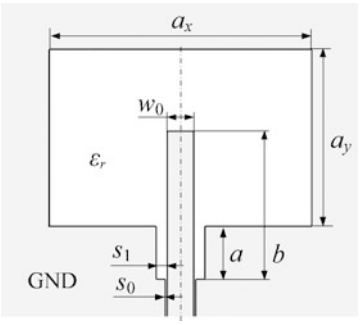


Fig. 2 Geometry of a UWB CPW-fed T-shaped slot antenna (Antenna 2, top view). The ground plane (GND) has infinite lateral extent

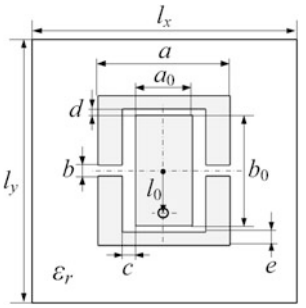


Fig. 3 Geometry of a broadband probe-fed microstrip patch antenna with two U-shaped parasitic elements (Antenna 3, top view). The dielectric substrate and ground plane both have lateral dimensions l_x and l_y. The *empty circle* below the center of the patch indicates the position of the feed pin

The training data were 400 geometries obtained by LHS, with 16 randomly selected frequencies per geometry ($n = 6{,}400$). The test data comprised 50 new LHS geometries with 121 equally spaced frequencies per geometry. We considered a high-fidelity model $\boldsymbol{R}_f$ (∼440,500 mesh cells, simulation time 12 min) and a low-fidelity model $\boldsymbol{R}_c$ (∼25,700 mesh cells, simulation time 15 s). It is instructive to consider three randomly picked responses from the training data, shown in Fig. 4. In spite of what appears to be a narrowly circumscribed input space (cf. the boundaries on the b, c, d, and e dimensions), the responses show considerable variety from one training point to the next. Furthermore, while within-training point coarse and fine responses agreed to some extent for some regions of the frequency band, there were considerable differences for others.

Surrogate models were constructed as before. The %RMSE values obtained with $\boldsymbol{R}_{s,\text{aux}}$ on the coarse test data, and $\boldsymbol{R}_s$ and $\boldsymbol{R}_{s,\text{full}}$ on the fine test data for $\beta \in \{0.15, 0.20, 0.25\}$ are shown in Table 1, as well as n_{SV}. The greatest data reduction, by 43 %, occurred for $\beta = 0.15$, while the %RMSE only increased from 5.53 % (full model) to 5.77 % (reduced model). The CPU time necessary to simu-

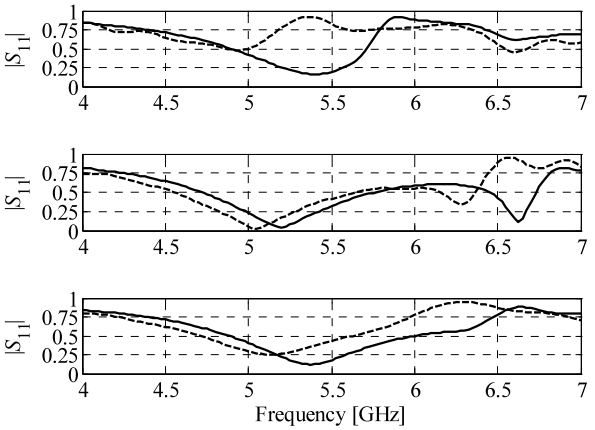


Fig. 4 Full-wave simulation responses $\boldsymbol{R}_c$ (- - -) and $\boldsymbol{R}_f$ (—) for the Antenna 3 training geometries $\boldsymbol{x} = [16.2267\ 0.7904\ 1.8943\ 0.4476\ 1.1148]^T$ mm (*top*), $[14.8396\ 0.6347\ 0.9242\ 0.9358\ 0.8196]^T$ mm (*center*), and $[15.2672\ 1.3194\ 1.2827\ 0.5165\ 1.0004]^T$ mm (*bottom*)

late fine-discretization training data for $\boldsymbol{R}_s$ for $\beta = 0.05$ (i.e., the model used for the optimization) was approximately 49 h; for $\boldsymbol{R}_{s,\text{full}}$ it was 80 h.

To further explore the influence of mesh density on our method, a second coarse model $\boldsymbol{R}_{cc}$ (~8,000 mesh cells, simulation time 8 s), i.e., coarser than $\boldsymbol{R}_c$, was generated, and corresponding surrogate models constructed. The predictive results for the new $\boldsymbol{R}_s$ were similar to previous results; e.g., for $\beta = 0.2$ the predictive %RMSE was 5.82 %, although the number of SVs increased somewhat to 3,992 (see Table 1). In order to evaluate the general similarity between the coarsely and finely simulated data, Pearson product-moment correlation coefficients were computed for the respective $|S_{11}|$ values, i.e., for $|S_{11}|$ of $\boldsymbol{R}_c$ and $\boldsymbol{R}_f$; and also for $\boldsymbol{R}_{cc}$ and $\boldsymbol{R}_f$ (using all training geometries with 121 equally spaced frequency points per geometry). The correlation coefficients were 0.74 and 0.51, respectively, suggesting some robustness to our procedure.

4.4 Application Examples: Antenna Optimization

The full and reduced BSVR models were used to perform design optimization of the antenna structures considered in Sects. 4.1 through 4.3. We again note that our models are intended as multipurpose global models that give accurate predictions for the whole of the input space; multiple optimization runs corresponding to any number of sets of design specifications constitute one kind of repeated-use application. The initial design in each case is the center of the region of interest $\boldsymbol{x}^{(0)}$. The design process starts by directly optimizing the BSVR model. Because of some limitations in the accuracy of the models given the design context (linear responses

were modeled—the preferred choice given the Gaussian kernel—but logarithmic responses (in decibels) are optimized), the design is further refined by means of the SM iterative process [14]

$$\boldsymbol{x}^{(i+1)} = \arg\min_{\boldsymbol{x}} U\left(\boldsymbol{R}_{\text{su}}^{(i)}(\boldsymbol{x})\right), \tag{8}$$

where $\boldsymbol{R}_{\text{su}}^{(i)}$ is a surrogate model, enhanced by frequency and output SM [14]. The surrogate model setup is performed using an evaluation of $\boldsymbol{R}_f$ at $\boldsymbol{x}^{(i)}$. U implements design specifications. For simplicity, we use the symbol $\boldsymbol{R}_{\text{co}}$ to denote either of $\boldsymbol{R}_{s.\text{full}}$ or $\boldsymbol{R}_s$, which can be considered the "coarse" models in the SM context. Let $\boldsymbol{R}_{\text{co}}(\boldsymbol{x}, F)$ denote the explicit dependency of the model on the frequency (F is the set of frequencies of interest at which the model is evaluated). The surrogate model is defined as

$$\boldsymbol{R}_{\text{su}}^{(i)}(\boldsymbol{x}) = \boldsymbol{R}_{\text{co}}\left(\boldsymbol{x}, \boldsymbol{\alpha}^{(i)} F\right) + \boldsymbol{d}^{(i)} \tag{9}$$

with

$$\boldsymbol{d}^{(i)} = \boldsymbol{R}_f\left(\boldsymbol{x}^{(i)}\right) - \boldsymbol{R}_{\text{co}}\left(\boldsymbol{x}^{(i)}, \boldsymbol{\alpha}^{(i)} F\right) \tag{10}$$

and

$$\boldsymbol{\alpha}^{(i)} F = \alpha_0^{(i)} + \alpha_1^{(i)} F \tag{11}$$

the affine frequency scaling (shift and scaling). The frequency scaling parameters are calculated as

$$\left[\alpha_0^{(i)} \alpha_1^{(i)}\right] = \arg\min_{\boldsymbol{x}} \left\| \boldsymbol{R}_f\left(\boldsymbol{x}^{(i)}\right) - \boldsymbol{R}_{\text{co}}\left(\boldsymbol{x}^{(i)}, \alpha_0^{(i)} + \alpha_1^{(i)} F\right) \right\| \tag{12}$$

i.e., to minimize the misalignment between the high-fidelity and the scaled low-fidelity model responses at $\boldsymbol{x}^{(i)}$. Although the models are evaluated at a discrete set of frequencies, the information at other frequencies can be obtained through interpolation. The misalignment is further reduced by the output SM (10); this ensures zero-order consistency (i.e., $\boldsymbol{R}_{\text{su}}^{(i)}(\boldsymbol{x}^{(i)}) = \boldsymbol{R}_f(\boldsymbol{x}^{(i)})$) between the surrogate and $\boldsymbol{R}_f$ [18]. The algorithm (8) working with the SM surrogate model (9)–(12) typically requires only three to four iterations to yield an optimized design, with the cost of each iteration effectively equal to a single evaluation of the high-fidelity model.

Figure 5 shows the responses of the reduced BSVR and fine models at the initial designs as well as the responses of the fine models at the final designs obtained for both antenna structures. The reduced BSVR models correspond to β values in Table 1 of 0.15 (Antenna 1), 0.05 (Antenna 2), and 0.15 (Antenna 3). Table 2 summarizes the results. One can see that the design quality and cost (expressed in terms of number of $\boldsymbol{R}_f$ evaluations) are very similar for the BSVR models obtained using full and reduced data sets (the CPU times associated with three $\boldsymbol{R}_f$ evaluations (Antennas 1 and 2) and four $\boldsymbol{R}_f$ evaluations (Antenna 3) were 36 min, 63 min, and 48 min, respectively).

For comparison, we also optimized the three antennas using a conventional (not surrogate-based) method, namely a state-of-the-art pattern search algorithm [19, 20] that directly relied on fine-discretization full-wave simulations (i.e., $\boldsymbol{R}_f$) for its objective function evaluations. While the maximum $|S_{11}|$ values at the final designs

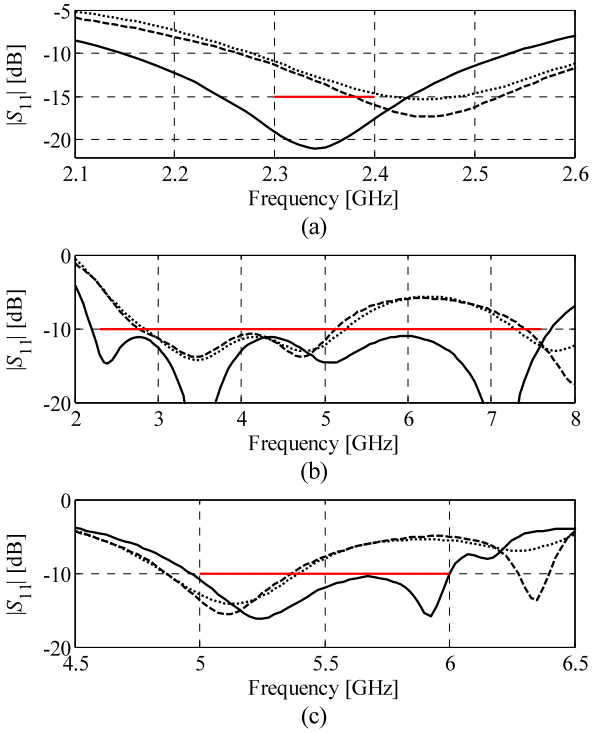


Fig. 5 Optimization results: responses of the BSVR model with reduced data set (· · ·), the fine model at the initial design (- - -), and the fine model at the optimized design (—) for (**a**) Antenna 1, (**b**) Antenna 2, and (**c**) Antenna 3. Design specifications are marked with *horizontal solid line*

Table 2 Results of antenna optimizations

Antenna	Model	max $\|S_{11}\|$ at final design[a]	Optimization cost[b]
1	$\boldsymbol{R}_{s,\text{full}}$	−18.3 dB	3
	$\boldsymbol{R}_s$	−17.6 dB	3
2	$\boldsymbol{R}_{s,\text{full}}$	−11.0 dB	3
	$\boldsymbol{R}_s$	−11.0 dB	3
3	$\boldsymbol{R}_{s,\text{full}}$	−10.2 dB	4
	$\boldsymbol{R}_s$	−10.2 dB	4

[a]max $|S_{11}|$ at the frequency band of interest: 2.3 to 2.4 GHz (Antenna 1), 2.3 to 7.7 GHz (Antenna 2), and 5 to 6 GHz (Antenna 3)

[b]Number of $\boldsymbol{R}_f$ evaluations including evaluation at the initial design

obtained for Antennas 1, 2, and 3 (−21.6 dB, −11.6 dB, and −10.7 dB, respectively) were similar to those obtained using our BSVR models and the above SM procedure, the computational expense for the conventional optimization was at least

an order of magnitude larger (i.e., 40, 148, and 201 $\boldsymbol{R}_f$ evaluations for Antennas 1, 2, and 3, respectively, compared to the 3, 3, and 4 $\boldsymbol{R}_f$ evaluations reported in Table 2). This accentuates how much faster optimization can be realized when accurate models such as BSVR models are available: our approach reduces by up to 43 % the high initial cost of setting up these multipurpose global models (in comparison the cost of the optimization using SM is insignificant).

5 Modeling and Optimization of Filters Using BSVR

Here we discuss global BSVR models based on reduced finely discretized data sets for the $|S_{21}|$ responses of two microstrip filters: a capacitively coupled dual-behavior resonator microstrip bandpass filter, and an open-loop ring resonator (OLRR) bandpass filter. As before, we use these models for design optimization.

5.1 Capacitively Coupled Dual-Behavior Resonator (CCDBR) Microstrip Bandpass Filter

Consider the capacitively coupled dual-behavior resonator (CCDBR) bandpass filter [21] implemented in microstrip lines, shown in Fig. 6(a). The three design variables were $\boldsymbol{x} = [L_1\ L_2\ L_3]^T$. The design variable space for the BSVR models was defined by the center vector $\boldsymbol{x}^0 = [3\ 5\ 1.5]^T$ mm and size vector $\boldsymbol{\delta} = [1\ 1\ 0.5]^T$ mm such that the variable ranges were $\boldsymbol{x}^0 \pm \boldsymbol{\delta}$ mm ($\boldsymbol{x}^0$ and $\boldsymbol{\delta}$ were guesses, guided to some extent by expert knowledge of the filters and a very rudimentary exploration of the design space). The substrate height was $h = 0.254$ mm and the relative permittivity was $\varepsilon_r = 9.9$; the value of S was 0.05 mm, while the microstrip line widths w_1 and w_2 were 0.25 mm and 0.5 mm, respectively. We were interested in the filter response over the frequency range 2 to 6 GHz. The high-fidelity model $\boldsymbol{R}_f$ of the filter was simulated using FEKO [22] (total mesh number 715, simulation time about 15 s per frequency). The low-fidelity model $\boldsymbol{R}_c$ was also simulated in FEKO (total mesh number 136, simulation time 0.6 s per frequency).

In order to set up the training data input vectors, 400 geometries were randomly selected from the design space using Latin hypercube sampling (LHS) [23]. For each geometry, 12 simulation frequencies were selected by uniform random sampling from the above frequency range, yielding a total of $n = 400 \times 12 = 4{,}800$ training input vectors of the form $\{\boldsymbol{x}_i = [L_{1i}\ L_{2i}\ L_{3i}\ f_i]^T | i = 1, \ldots, n\}$, with L_{1i}, L_{2i}, and L_{3i} the design variables corresponding to the ith input vector, and f_i a frequency value within the range of interest. The corresponding output scalars, obtained from FEKO simulations, were $y_i = |S_{21i}|$. The test data were likewise obtained from 50 new geometries, also obtained via LHS, with 41 equally spaced frequencies per geometry. The training data were simulated at the $\boldsymbol{R}_c$ mesh density and used to train the BSVR model $\boldsymbol{R}_{s,\text{aux}}$ for $\beta = 0.1$, 0.2, and 0.3 (β is the loss function parameter

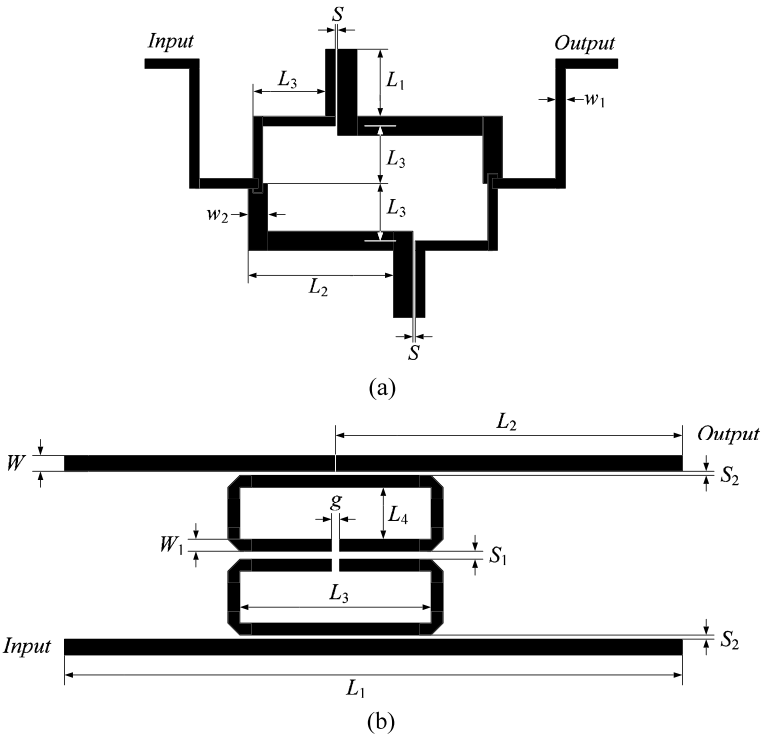


Fig. 6 Geometry of the (**a**) CCDBR bandpass filter [21] and (**b**) OLRR bandpass filter [24]

in Eq. (3) discussed above). $\boldsymbol{R}_{s,\text{aux}}$ was used to make predictions on the test data (also simulated at the $\boldsymbol{R}_c$ mesh density).

The %RMSE (root mean square error normalized to the target range expressed as a percentage) was around 4.13 % for the three β values, which was acceptable for this highly nonlinear problem, and indicated that the training set was sufficiently large. Next, for each value of β the n_{SV} SVs of $\boldsymbol{R}_{s,\text{aux}}$ were simulated at the $\boldsymbol{R}_f$ mesh density. BSVR models fitted to these reduced training sets gave the desired surrogate models $\boldsymbol{R}_s$.

For comparison, surrogates $\boldsymbol{R}_{s,\text{full}}$ trained on the full fine-discretization training data ($n = 4{,}800$) were also set up. Table 3 gives, for the three β values, the %RMSEs obtained with $\boldsymbol{R}_{s,\text{aux}}$ on the coarse test data, and with $\boldsymbol{R}_s$ and $\boldsymbol{R}_{s,\text{full}}$ on the fine test data; as well as the number of SVs obtained in each instance. The highly similar %RMSEs obtained with $\boldsymbol{R}_s$ and $\boldsymbol{R}_{s,\text{full}}$ indicate that reducing the number of expensive fine-discretization training points from n to n_{SV} incurred negligible accuracy loss, even for a reduction in data as large as 48 % ($\beta = 0.1$). Figure 7(a) shows typical predictive results for $|S_{21}|$ versus frequency obtained for the test geometry $\boldsymbol{x} = [2.794\ 4.407\ 1.491]^T$ mm. Some discrepancy can be observed when comparing the RMSE values for $\boldsymbol{R}_{s,\text{aux}}$ in Table 3 to those for $\boldsymbol{R}_s$ (and $\boldsymbol{R}_{s,\text{full}}$). This occurs because the coarse model responses are slightly smoother as functions of fre-

Table 3 Predictive errors of surrogate filter models. $\boldsymbol{R}_{s,\text{aux}}$ are BSVR models trained on all n data points obtained from coarsely discretized simulations, $\boldsymbol{R}_{s,\text{full}}$ are BSVR models trained on all n data points obtained from finely discretized simulations, and $\boldsymbol{R}_s$ are BSVR models trained on the n_{SV} finely discretized data points corresponding to SVs of $\boldsymbol{R}_{s,\text{aux}}$

Filter	β	RMSE (%)			n_{SV}	n_{SV}/n (%)
		$\boldsymbol{R}_{s,\text{aux}}$	$\boldsymbol{R}_s$	$\boldsymbol{R}_{s,\text{full}}$		
CCDBR ($n = 4{,}800$)	0.1	4.17	6.05	5.98	2,488	52
	0.2	4.12	5.87	5.95	2,563	53
	0.3	4.11	5.95	5.86	2,886	60
OLRR ($n = 4{,}800$)	0.1	4.12	4.26	4.21	2,360	49
	0.2	3.77	4.47	4.32	2,565	53
	0.3	3.46	3.90	3.64	2,744	57

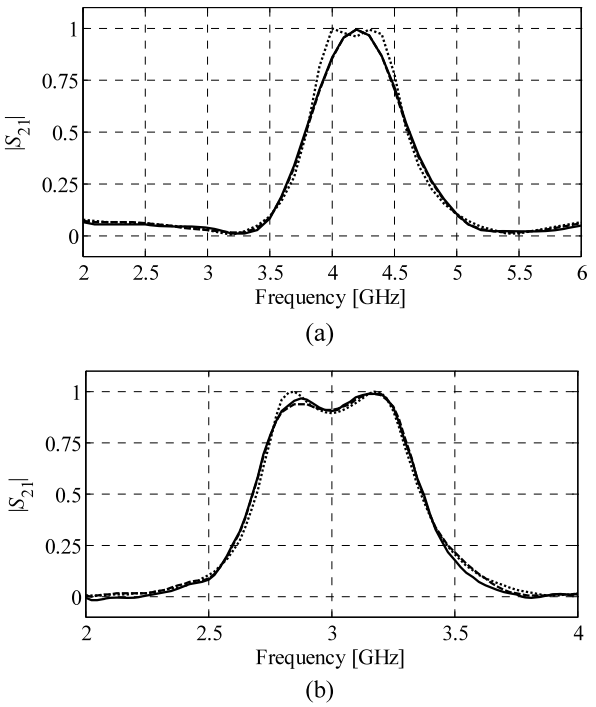


Fig. 7 Predictive response of BSVR models $\boldsymbol{R}_s$ (—) and $\boldsymbol{R}_{s,\text{full}}$ (- - -), and high-fidelity model $\boldsymbol{R}_f$ ($\cdots$) for (**a**) the CCDBR bandpass filter test geometry $\boldsymbol{x} = [2.794\ 4.407\ 1.491]^T$ mm, and (**b**) the OLRR bandpass filter test geometry $\boldsymbol{x} = [38.088\ 8.306\ 5.882\ 4.029\ 0.193\ 0.061\ 0.985]^T$ mm

quency (i.e., they do not contain as much detail particularly in the passband) than the fine model ones, which make them easier to model; this results in a lower value of %RMSE. Using a finer mesh for the $\boldsymbol{R}_c$ model would reduce this discrepancy. The

RMSE values for $\boldsymbol{R}_s$ and $\boldsymbol{R}_{s.\text{full}}$ nevertheless were good given the highly nonlinear nature of the modeling problem, and of sufficient accuracy to yield good optimization results, as we show in Sect. 4. The total computational time necessary to gather the training data for setting up $\boldsymbol{R}_{s.\text{full}}$ was 20 h, whereas the corresponding time for setting $\boldsymbol{R}_s$ (including both low- and high-fidelity model evaluations) was 11.2, 11.4, and 12.8 h (for $\beta = 0.1$, 0.2, and 0.3, respectively) on a quad core PC with a 2.66 GHz Intel processor and 4 GB RAM. Thus the computational savings due to the proposed technique vary between 36 % (for $\beta = 0.3$) to 44 % (for $\beta = 0.1$).

5.2 Open-Loop Ring Resonator (OLRR) Bandpass Filter

The filter geometry [24] is shown in Fig. 6(b). The seven design variables were $\boldsymbol{x} = [L_1\ L_2\ L_3\ L_4\ S_1\ S_2\ G]^T$. The design space was described by $\boldsymbol{x}^0 = [40\ 8\ 6\ 4\ 0.2\ 0.1\ 1]^T$ mm and $\boldsymbol{\delta} = [2\ 1\ 0.4\ 0.4\ 0.1\ 0.05\ 0.2]^T$ mm. The substrate parameters were $h = 0.635$ mm and $\varepsilon_r = 10.2$, while the microstrip line widths were $W_1 = 0.4$ mm and $W = 0.6$ mm. The frequency range of interest was 2 to 4 GHz. High- and low-fidelity models were simulated in FEKO [22] (total mesh number 1,084 and simulation time 40 s per frequency for $\boldsymbol{R}_f$; total mesh number 148 and simulation time 0.8 s per frequency for $\boldsymbol{R}_c$). The training data comprised 400 geometries obtained by LHS [23], with 12 randomly selected frequencies per geometry ($n = 4{,}800$), while the test data were 50 new LHS geometries with 81 equally spaced frequencies per geometry. Setting up $\boldsymbol{R}_{s,\text{aux}}$, $\boldsymbol{R}_s$, and $\boldsymbol{R}_{s,\text{full}}$ proceeded in a manner similar to the earlier filter. Table 3 gives the %RMSE values obtained with $\boldsymbol{R}_{s,\text{aux}}$ on the coarse test data and with $\boldsymbol{R}_s$ and $\boldsymbol{R}_{s,\text{full}}$ on the fine test data; as well as the SV counts. In general, %RMSE values of $\boldsymbol{R}_s$ were only marginally higher than those of $\boldsymbol{R}_{s,\text{full}}$, suggesting as before that reducing the number of training points from n to n_{SV} by using the SVs of $\boldsymbol{R}_{s,\text{aux}}$ as fine-discretization training points for $\boldsymbol{R}_s$ has little effect on the prediction accuracy. The greatest reduction in data (51 %) was obtained for $\beta = 0.1$. Figure 7(b) shows representative predictive results for $|S_{21}|$ versus frequency, in particular for the test geometry $\boldsymbol{x} = [38.088\ 8.306\ 5.882\ 4.029\ 0.193\ 0.061\ 0.985]^T$ mm. The computational time necessary to gather the training data for setting up $\boldsymbol{R}_{s.\text{full}}$ was 53.3 h. The corresponding time for setting up $\boldsymbol{R}_s$ (including both low- and high-fidelity model evaluations) was 27.2, 29.3, and 31.4 h ($\beta = 0.1$, 0.2, and 0.3, respectively). The computational savings due to the proposed technique range from 41 % (for $\beta = 0.3$) to 49 % (for $\beta = 0.1$).

5.3 Filter Optimization Using BSVR Surrogates

The BSVR developed in Sects. 5.1 and 5.2 is used to perform design optimization of the filters. The optimization methodology is essentially the same as described in

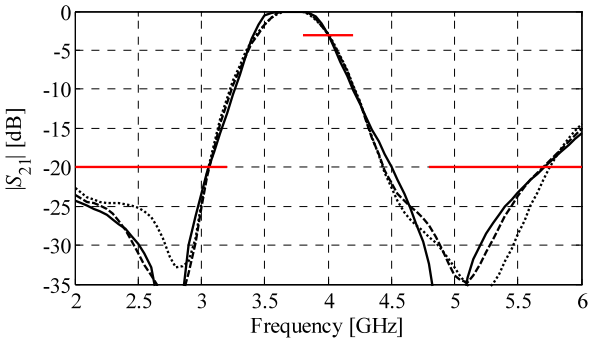


Fig. 8 CCDBR filter: responses of the high-fidelity model (—) and BSVR models $\boldsymbol{R}_{s.\text{full}}$ (- - -) and $\boldsymbol{R}_s$ ($\cdots$) at the initial design $\boldsymbol{x}^{(0)}$. Design specification as per *horizontal solid* (*red*) *lines*

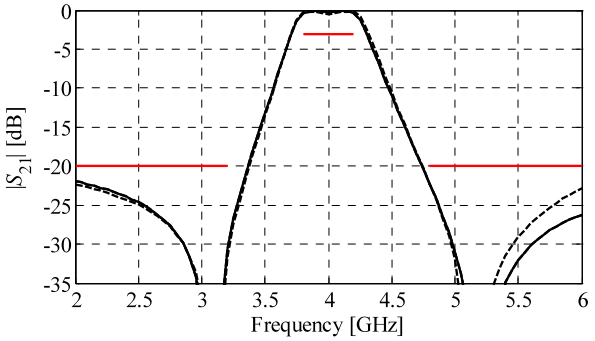


Fig. 9 CCDBR filter: responses of the high-fidelity model at the optimized designs found using $\boldsymbol{R}_{s.\text{full}}$ (—) and $\boldsymbol{R}_s$ (- - -). Design specification as per *horizontal solid* (*red*) *lines*

Sect. 4.4 and involves iterative correction and optimization of the surrogates in order to (locally) improve their accuracy [25].

The design process starts by directly optimizing the BSVR model (for each filter, we used the BSVR surrogates corresponding to $\beta = 0.1$). Each iteration (see Eq. (8)) requires only one evaluation of the high-fidelity model. The CCDBR bandpass filter had design specifications $|S_{21}| \geq -3$ dB for $3.8 \leq f \leq 4.2$ GHz; and $|S_{21}| \leq -20$ dB for $2.0 \leq f \leq 3.2$ GHz, and $4.8 \leq f \leq 6.0$ GHz (f denotes frequency). Figure 8 shows the responses of the high-fidelity model $\boldsymbol{R}_f$ as well as the responses of $\boldsymbol{R}_s$ and $\boldsymbol{R}_{s.\text{full}}$ at the initial design $\boldsymbol{x}^{(0)} = [3\ 6\ 1.5]^T$ mm. The high-fidelity model responses at the optimized designs found using both $\boldsymbol{R}_s$ and $\boldsymbol{R}_{s.\text{full}}$ are shown in Fig. 9 (these designs were $[3.47\ 4.79\ 1.01]^T$ mm and $[3.21\ 4.87\ 1.22]^T$ mm, respectively). In both cases, the design process is accomplished in three iterations, which correspond to a design cost of four high-fidelity model evaluations.

For the OLRR bandpass filter, the design specifications were $|S_{21}| \geq -1$ dB for $2.85 \leq f \leq 3.15$ GHz, and $|S_{21}| \leq -20$ dB for $2.0 \leq f \leq 2.5$ GHz and

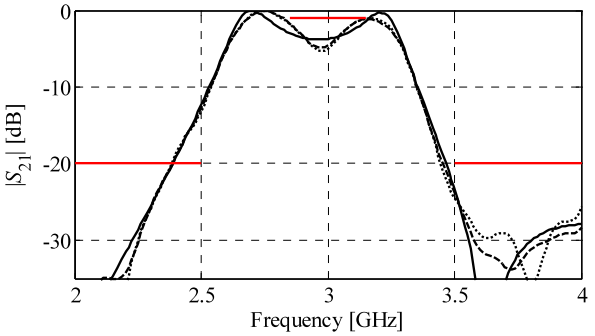


Fig. 10 OLRR bandpass filter: responses of the high-fidelity model (—) and BSVR models $\boldsymbol{R}_{s.\text{full}}$ (- - -) and $\boldsymbol{R}_s$ ($\cdots$) at the initial design $\boldsymbol{x}^{(0)}$. Design specification as per *horizontal solid* (*red*) *lines*

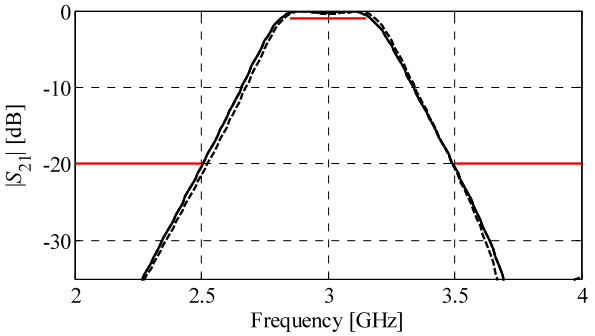


Fig. 11 OLRR bandpass filter: responses of the high-fidelity model at the optimized designs found using $\boldsymbol{R}_{s.\text{full}}$ (—) and $\boldsymbol{R}_s$ (- - -). Design specification as per *horizontal solid* (*red*) *lines*

$3.5 \leq f \leq 4.0$ GHz. Figure 10 shows the responses of $\boldsymbol{R}_f$, $\boldsymbol{R}_s$, and $\boldsymbol{R}_{s.\text{full}}$ at the initial design $\boldsymbol{x}^{(0)} = [40.0\ 8.0\ 6.0\ 4.0\ 0.1\ 0.1\ 1.0]^T$ mm. The responses at the optimized designs obtained using $\boldsymbol{R}_s$ and $\boldsymbol{R}_{s.\text{full}}$ are shown in Fig. 11 (these designs were $[39.605\ 8.619\ 6.369\ 3.718\ 0.300\ 0.069\ 0.986]^T$ mm and $[39.010\ 8.219\ 5.786\ 4.260\ 0.268\ 0.050\ 1.068]^T$ mm, respectively). In both cases, the design process is accomplished in two iterations, which correspond to the design cost of three high-fidelity model evaluations.

6 Variable-Fidelity Optimization Using Local BSVR Surrogates

In this section, we discuss the application of BSVR surrogates defined locally (i.e., in a specific region of the input space) towards low-cost design optimization of antennas [26].

As we have shown, BSVR models can be accurate; however, similarly to other approximation-based modeling methods, considerable computational overhead is necessary to acquire the training data. This is not convenient when using approximation surrogates for *ad hoc* optimization of a specific structure.

Here, we describe a computationally efficient antenna design methodology that combines space mapping (as the optimization engine), and coarse model response surface approximation implemented through BSVR. BSVR serves to create a fast coarse model of the antenna structure. In order to reduce the computational cost of creating the latter, it is obtained from coarse discretization EM simulation data.

6.1 Optimization Algorithm

As mentioned before, the main optimization engine is space mapping (SM) [26]. The generic SM optimization algorithm produces a sequence of approximate solutions to the problem (1), $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \ldots$, as follows:

$$\boldsymbol{x}^{(i+1)} = \arg\min_{\boldsymbol{x}} U\left(\boldsymbol{R}_s^{(i)}(\boldsymbol{x})\right), \tag{13}$$

where $\boldsymbol{R}_s^{(i)}$ is the SM surrogate model at iteration i. $\boldsymbol{x}^{(0)}$ is the initial design. The surrogate model is constructed using the underlying coarse model $\boldsymbol{R}_c$ and suitable auxiliary transformations [27]. The elementary SM transformations include input SM [28] with the surrogate defined as $\boldsymbol{R}_s^{(i)}(\boldsymbol{x}) = \boldsymbol{R}_c(\boldsymbol{x} + \boldsymbol{c}^{(i)})$, multiplicative output SM [27], $\boldsymbol{R}_s^{(i)}(\boldsymbol{x}) = \boldsymbol{A}^{(i)} \cdot \boldsymbol{R}_c(\boldsymbol{x})$, additive output SM [28], $\boldsymbol{R}_s^{(i)}(\boldsymbol{x}) = \boldsymbol{R}_c(\boldsymbol{x}) + \boldsymbol{d}^{(i)}$, and frequency scaling [27], $\boldsymbol{R}_s^{(i)}(\boldsymbol{x}) = \boldsymbol{R}_{c.f}(\boldsymbol{x}; \boldsymbol{F}^{(i)})$. In frequency SM, it is assumed that the coarse model is an evaluation of a given performance parameter over a certain frequency range, i.e., $\boldsymbol{R}_c(\boldsymbol{x}) = [R_c(\boldsymbol{x}; \omega_1) \ldots R_c(\boldsymbol{x}; \omega_m)]^T$, and the frequency-scaled model is then given as $\boldsymbol{R}_{c.f}(\boldsymbol{x}; \boldsymbol{F}^{(i)}) = [R_c(\boldsymbol{x}; s(\omega_1)) \ldots R_c(\boldsymbol{x}; s(\omega_m))]^T$, where s is a scaling function dependent on the set of parameters $\boldsymbol{F}^{(i)}$. Typically, a linear scaling function $s(\omega) = f_0^{(i)} + f_1^{(i)}\omega$ is used.

Parameters of SM transformations are obtained using the parameter extraction (PE) process, which, in the case of input SM, takes the form

$$\boldsymbol{c}^{(i)} = \arg\min_{\boldsymbol{c}} \sum_{k=0}^{i} \left\| \boldsymbol{R}_f\left(\boldsymbol{x}^{(k)}\right) - \boldsymbol{R}_c\left(\boldsymbol{x}^{(k)} + \boldsymbol{c}\right) \right\|. \tag{14}$$

Formulation of PE for the other transformations is similar [28].

Because PE and surrogate model optimization may require a large number of coarse model evaluations, it is beneficial that $\boldsymbol{R}_c$ is fast, which is usually not possible for antenna structures, where the only universally available (and yet accurate) type of coarse model is the output of coarse-discretization EM simulations. To alleviate this problem, we construct the coarse model by using a fixed number of such low-fidelity simulations as training data for the coarse model, so that, once set up, the coarse model can be used by the SM algorithm without further reference to the EM solver.

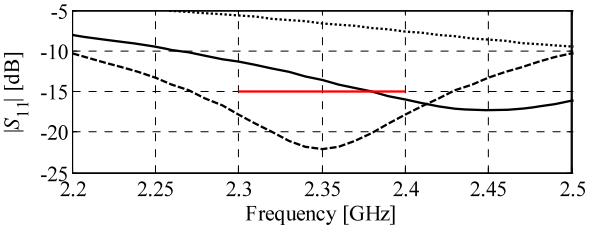


Fig. 12 CPW-fed slot dipole antenna: responses of the low- (· · · ·) and high- (—) fidelity models at the initial design $\boldsymbol{x}^{\text{init}}$, and the response of the low-fidelity model at its approximate optimum $\boldsymbol{x}^{(0)}$ (- - -)

In order to improve the convergence properties of the algorithm, it is embedded in the trust region framework [29], so that the new design $\boldsymbol{x}^{(i+1)}$ is found only in the vicinity of the current one, $\boldsymbol{x}^{(i)}$, as follows:

$$\boldsymbol{x}^{(i+1)} = \arg \min_{\boldsymbol{x}, \|\boldsymbol{x}-\boldsymbol{x}^{(i)}\| \le \delta^{(i)}} U\left(\boldsymbol{R}_s^{(i)}(\boldsymbol{x})\right), \tag{15}$$

where $\delta^{(i)}$ is the trust region radius updated in each iteration according to the standard rules [29]. Within this framework, the designs that do not improve the specification error are rejected, and the search is repeated with the reduced value of $\delta^{(i)}$.

The steps in the modeling procedure are as follows. First, we find an approximate optimum of the coarse model $\boldsymbol{R}_{\text{cd}}$ (i.e., the low-fidelity full-wave EM simulations). Then we construct a BSVR surrogate $\boldsymbol{R}_c$, only using relatively densely spaced training data in the vicinity of this optimum, thus further enhancing the computational efficiency. Once constructed, $\boldsymbol{R}_c$ is used as the basis for the iterative optimization (13)–(15).

6.2 Antenna Optimization Examples

As examples, we consider again the CPW-fed slot dipole antenna of Sect. 4.1 (Antenna 1, Fig. 1) and the CPW-fed T-shaped slot antenna of Sect. 4.2 (Antenna 2, Fig. 2).

For Antenna 1, we have two design variables, $\boldsymbol{x} = [W \ L]^T$ mm. Using CST Microwave Studio [15], we consider a high-fidelity model $\boldsymbol{R}_f$ (∼130,000 mesh cells, simulation time 12 min), and a low-fidelity model $\boldsymbol{R}_{\text{cd}}$ (∼5,000 mesh cells, simulation time 30 s). The initial design is $\boldsymbol{x}^{\text{init}} = [7.5 \ 39.0]^T$ mm. The starting point of the SM optimization is the approximate optimum of the low-fidelity model, $\boldsymbol{x}^{(0)} = [5.0 \ 43.36]^T$ mm, found using a pattern search algorithm [19]. The computational cost of this step was 26 evaluations of $\boldsymbol{R}_{\text{cd}}$. The BSVR coarse model has been created using 100 low-fidelity model samples allocated using LHS [23] in the vicinity of $\boldsymbol{x}^{(0)}$ defined by deviation $\boldsymbol{d} = [1 \ 3]^T$ mm. The size of this vicinity must be sufficiently large to allow the coarse model to "absorb" the misalignment between the low- and high-fidelity models at $\boldsymbol{x}^{(0)}$ through appropriate SM transformations.

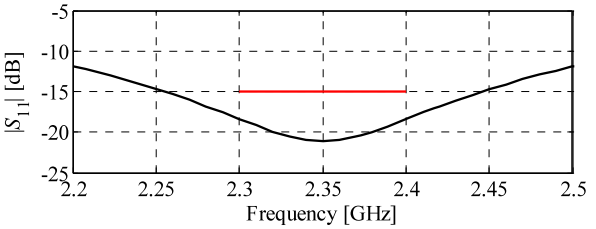


Fig. 13 CPW-fed slot dipole antenna: response of the high-fidelity model at the final design $\boldsymbol{x}^{(4)}$

Table 4 CPW-fed slot dipole antenna: design cost

Algorithm component	Number of model evaluations	Evaluation time	
		Absolute [min]	Relative to $\boldsymbol{R}_f$
Optimization of $\boldsymbol{R}_{\text{cd}}$	$26 \times \boldsymbol{R}_{\text{cd}}$	13	1.1
Creating BSVR coarse model	$100 \times \boldsymbol{R}_{\text{cd}}$	50	4.2
Evaluation of $\boldsymbol{R}_f$[a]	$5 \times \boldsymbol{R}_f$	60	5.0
Total optimization time	N/A	123	10.3

[a]Includes evaluation of $\boldsymbol{R}_f$ at $\boldsymbol{x}^{(0)}$

The low-fidelity model at the initial design, as well as the low- and high-fidelity model responses at $\boldsymbol{x}^{(0)}$, are shown in Fig. 12. In this case, the major discrepancy between the models is a frequency shift. Therefore, the primary SM transformation used for this example is frequency scaling applied to all designs, $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(i)}$, considered during the optimization run. The SM surrogate is then enhanced using a local additive output SM [27] so that the entire SM model has the form $\boldsymbol{R}_s^{(i)}(\boldsymbol{x}) = \boldsymbol{R}_{c.f}(\boldsymbol{x}; \boldsymbol{F}^{(i)}) + [\boldsymbol{R}_f(\boldsymbol{x}^{(i)}) - \boldsymbol{R}_c(\boldsymbol{x}^{(i)})]$.

The final design, $\boldsymbol{x}^{(4)} = [5.00\ 41.56]^T$ mm, is obtained in four SM iterations. The high-fidelity model response at $\boldsymbol{x}^{(4)}$ is shown in Fig. 13. At this design, we have $|S_{11}| \leq -18.3$ dB over the entire frequency band of interest. The design cost is summarized in Table 4 and corresponds to about ten evaluations of the high-fidelity model. Figure 14 shows the convergence plot and the evolution of the specification error versus iteration index.

For Antenna 2, the design variables are $\boldsymbol{x} = [a_x\ a_y\ a\ b]^T$ mm. The design specifications are $|S_{11}| \leq -12$ dB for 2.3 to 7.6 GHz. The high-fidelity model $\boldsymbol{R}_f$ is evaluated with the CST MWS transient solver [15] (3,556,224 mesh cells, simulated in 60 min). The low-fidelity model $\boldsymbol{R}_{\text{cd}}$ is also evaluated in CST MWS but with a coarser mesh: 110,208 mesh cells, simulation time 1.5 min. The initial design is $\boldsymbol{x}^{\text{init}} = [40\ 30\ 10\ 20]^T$ mm. The approximate low-fidelity model optimum, $\boldsymbol{x}^{(0)} = [40.33\ 25.6\ 8.4\ 20.8]^T$ mm, has been found using a pattern search algorithm, at the cost of 85 evaluations of $\boldsymbol{R}_{\text{cd}}$. The BSVR coarse model has been created using 100 low-fidelity model samples allocated using LHS in the vicinity of $\boldsymbol{x}^{(0)}$ defined by deviation $\boldsymbol{d} = [2\ 2\ 1\ 1]^T$ mm. The BSVR model $\boldsymbol{R}_c$ was subsequently used as a

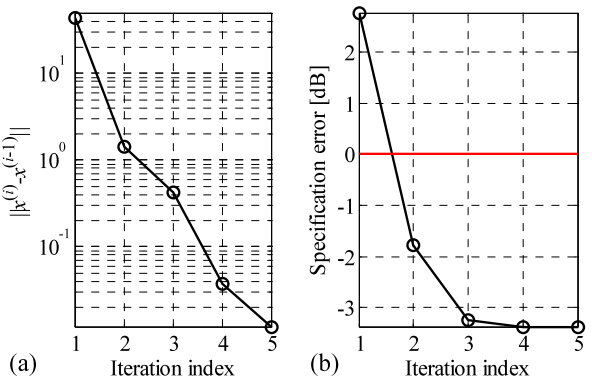


Fig. 14 CPW-fed slot dipole antenna: (**a**) convergence plot, (**b**) specification error versus iteration index

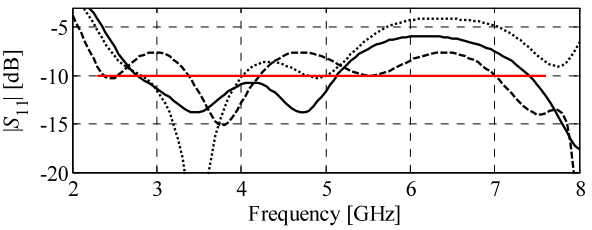


Fig. 15 Broadband CPW-fed T-shaped slot antenna: responses of the low- (· · · ·) and high- (—) fidelity models at the initial design $\boldsymbol{x}^{\text{init}}$ and the response of the low-fidelity model at its approximate optimum $\boldsymbol{x}^{(0)}$ (- - -)

coarse model for the SM algorithm. Figure 15 shows the low-fidelity model at the initial design, as well as the low- and high-fidelity model responses at $\boldsymbol{x}^{(0)}$. Because the major discrepancy between the models is a vertical shift, the primary SM transformation used for this example is the multiplicative response correction applied to all designs, $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(i)}$, considered during the optimization run. The SM surrogate is then enhanced using an additive output SM with the SM model having the form $\boldsymbol{R}_s^{(i)}(\boldsymbol{x}) = \boldsymbol{A}^{(i)} \cdot \boldsymbol{R}_c(\boldsymbol{x}) + [\boldsymbol{R}_f(\boldsymbol{x}^{(i)}) - \boldsymbol{R}_c(\boldsymbol{x}^{(i)})]$.

The final design, $\boldsymbol{x}^{(7)} = [39.84\ 24.52\ 8.84\ 21.40]^T$ mm, is obtained in seven SM iterations. The high-fidelity model response at $\boldsymbol{x}^{(7)}$ is shown in Fig. 16. At that design, we have $|S_{11}| \leq -10.9$ dB for 2.3 GHz to 7.6 GHz. The overall design cost is summarized in Table 5 and corresponds to about 15 evaluations of the high-fidelity model. The convergence of the algorithm (Fig. 17) is consistent with that for the previous example.

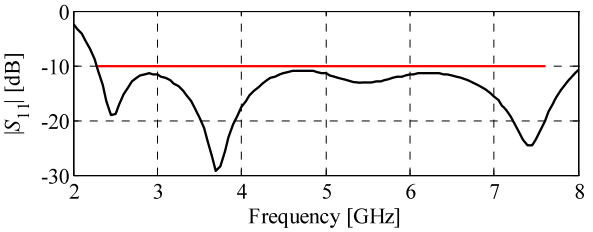


Fig. 16 Broadband CPW-fed T-shaped slot antenna: response of the high-fidelity model at the final design $\boldsymbol{x}^{(7)}$

Table 5 Broadband CPW-fed T-shaped slot antenna: design cost

Algorithm component	Number of model evaluations	Evaluation time	
		Absolute [min]	Relative to $\boldsymbol{R}_f$
Optimization of $\boldsymbol{R}_{\text{cd}}$	$85 \times \boldsymbol{R}_{\text{cd}}$	128	2.1
Creating BSVR coarse model	$100 \times \boldsymbol{R}_{\text{cd}}$	150	2.5
Evaluation of $\boldsymbol{R}_f$ [a]	$10 \times \boldsymbol{R}_f$	600	10.0
Total optimization time	N/A	878	14.6

[a]Includes evaluation of $\boldsymbol{R}_f$ at $\boldsymbol{x}^{(0)}$

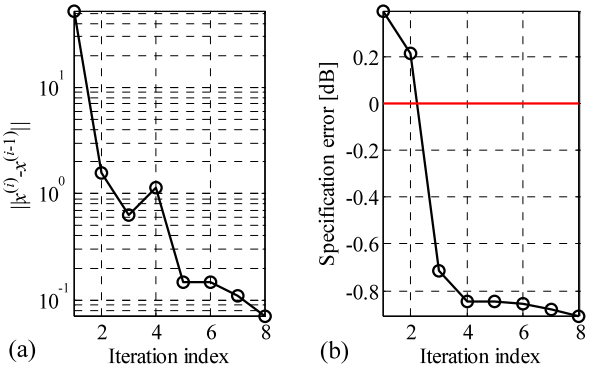


Fig. 17 Broadband CPW-fed T-shaped slot antenna: (**a**) convergence plot, (**b**) specification error versus iteration index

7 Conclusion

In this chapter, we presented a Bayesian support vector regression methodology for accurate modeling of microwave components and structures. We demonstrated the possibility of reduction of the number of fine-discretization training points by performing BSVR modeling on coarse-discretization EM simulation data (selected by standard experimental design) and then obtaining high-fidelity simulations only for

the points that contribute to the initial BSVR model in a nontrivial way. The computational savings thus obtained had little effect on the modeling accuracy. We have also demonstrated that the reduced-training-set BSVR models perform as well as the full-training-set models in parametric optimization of antenna structures. A notable advantage of BSVR is that only a single parameter must be set by the user, namely β (hyperparameters are initialized randomly during training). This is in contrast to, for instance, neural network-based methodologies for regression, which might require the tuning of a variety of architectural/performance parameters (e.g., number of hidden units, learning rate, momentum). We also discussed the use of BSVR surrogates for variable-fidelity design optimization of antennas, where the main optimization engine is space mapping, whereas the underlying coarse model is obtained by approximating low-fidelity EM simulation data. As a result, the optimization can be accomplished at a low computational cost corresponding to a few evaluations of the high-fidelity EM simulations of the structure under design.

References

1. Angiulli, G., Cacciola, M., Versaci, M.: Microwave devices and antennas modelling by support vector regression machines. IEEE Trans. Magn. **43**, 1589–1592 (2007)
2. Jacobs, J.P.: Bayesian support vector regression with automatic relevance determination kernel for modeling of antenna input characteristics. IEEE Trans. Antennas Propag. **60**, 2114–2118 (2012)
3. Chu, W., Keerthi, S.S., Ong, C.J.: Bayesian support vector regression using a unified loss function. IEEE Trans. Neural Netw. **15**, 29–44 (2004)
4. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2006)
5. Devabhaktuni, V.K., Yagoub, M.C.E., Zhang, Q.J.: A robust algorithm for automatic development of neural network models for microwave applications. IEEE Trans. Microw. Theory Tech. **49**, 2282–2291 (2001)
6. Couckuyt, I., Declercq, F., Dhaene, T., Rogier, H., Knockaert, L.: Surrogate-based infill optimization applied to electromagnetic problems. Int. J. RF Microw. Comput.-Aided Eng. **20**, 492–501 (2010)
7. Tokan, N.T., Gunes, F.: Knowledge-based support vector synthesis of the microstrip lines. Prog. Electromagn. Res. **92**, 65–77 (2009)
8. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2002)
9. Bandler, J.W., Georgieva, N., Ismail, M.A., Rayas-Sánchez, J.E., Zhang, Q.J.: A generalized space mapping tableau approach to device modeling. IEEE Trans. Microw. Theory Tech. **49**, 67–79 (2001)
10. Rayas-Sánchez, E., Gutierrez-Ayala, V.: EM-based Monte Carlo analysis and yield prediction of microwave circuits using linear-input neural-output space mapping. IEEE Trans. Microw. Theory Tech. **54**, 4528–4537 (2006)
11. Koziel, S., Bandler, J.W.: Recent advances in space-mapping-based modeling of microwave devices. Int. J. Numer. Model. **23**, 425–446 (2010)
12. Zhang, L., Zhang, Q.J., Wood, J.: Statistical neuro-space mapping technique for large-signal modeling of nonlinear devices. IEEE Trans. Microw. Theory Tech. **56**, 2453–2467 (2011)
13. Bandler, J.W., Cheng, Q.S., Koziel, S.: Simplified space mapping approach to enhancement of microwave device models. Int. J. RF Microw. Comput.-Aided Eng. **16**, 518–535 (2006)

14. Koziel, S., Bandler, S.W., Madsen, K.: A space mapping framework for engineering optimization: theory and implementation. IEEE Trans. Microw. Theory Tech. **54**, 3721–3730 (2006)
15. CST Microwave Studio, ver. 2011. CST AG, Bad Nauheimer Str. 19, D-64289 Darmstadt, Germany (2012)
16. Jiao, J.-J., Zhao, G., Zhang, F.-S., Yuan, H.-W., Jiao, Y.-C.: A broadband CPW-fed T-shape slot antenna. Prog. Electromagn. Res. **76**, 237–242 (2007)
17. Wi, S.-H., Lee, Y.-S., Yook, J.-G.: Wideband microstrip patch antenna with U-shaped parasitic elements. IEEE Trans. Antennas Propag. **55**, 1196–1199 (2007)
18. Alexandrov, N.M., Lewis, R.M.: An overview of first-order model management for engineering optimization. Optim. Eng. **2**, 413–430 (2001)
19. Kolda, T.G., Lewis, R.M., Torczon, V.: Optimization by direct search: new perspectives on some classical and modern methods. SIAM Rev. **45**, 385–482 (2003)
20. Koziel, S.: Multi-fidelity multi-grid design optimization of planar microwave structures with Sonnet. In: International Review of Progress in Applied Computational Electromagnetics, April 26–29, Tampere, Finland, pp. 719–724 (2010)
21. Manchec, A., Quendo, C., Favennec, J.-F., Rius, E., Person, C.: Synthesis of capacitive-coupled dual-behavior resonator (CCDBR) filters. IEEE Trans. Microw. Theory Tech. **54**, 2346–2355 (2006)
22. FEKO® User's Manual, Suite 6.0. EM Software & Systems-S.A. (Pty) Ltd, 32 Techno Lane, Technopark, Stellenbosch, 7600, South Africa (2010)
23. Beachkofski, B., Grandhi, R.: Improved distributed hypercube sampling. American Institute of Aeronautics and Astronautics. Paper AIAA 2002-1274 (2002)
24. Chen, C.Y., Hsu, C.Y.: A simple and effective method for microstrip dual-band filters design. IEEE Microw. Wirel. Compon. Lett. **16**, 246–248 (2006)
25. Koziel, S., Echeverría-Ciaurri, D., Leifsson, L.: Surrogate-based methods. In: Koziel, S., Yang, X.S. (eds.) Computational Optimization, Methods and Algorithms. Studies in Computational Intelligence, pp. 33–60. Springer, Berlin (2011)
26. Koziel, S., Ogurtsov, S., Jacobs, J.P.: Low-cost design optimization of slot antennas using Bayesian support vector regression and space mapping. In: Loughborough Antennas and Propagation Conf. (2012). doi:10.1109/LAPC.2012.6402988
27. Koziel, S., Cheng, Q.S., Bandler, J.W.: Space mapping. IEEE Microw. Mag. **9**, 105–122 (2008)
28. Bandler, J.W., Cheng, Q.S., Dakroury, S.A., Mohamed, A.S., Bakr, M.H., Madsen, K., Sondergaard, J.: Space mapping: the state of the art. IEEE Trans. Microw. Theory Tech. **52**, 337–361 (2004)
29. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust Region Methods. MPS-SIAM Series on Optimization. Springer, Berlin (2000)

Artificial Neural Networks and Space Mapping for EM-Based Modeling and Design of Microwave Circuits

José Ernesto Rayas-Sánchez

Abstract This chapter reviews the intersection of two major CAD technologies for modeling and design of RF and microwave circuits: artificial neural networks (ANNs) and space mapping (SM). A brief introduction to ANNs is first presented, starting from elementary concepts associated to biological neurons. Electromagnetic (EM)-based modeling and design optimization of microwave circuits using ANNs is addressed. The conventional and most widely used neural network approach for RF and microwave design optimization is explained, followed by brief descriptions of typical enhancing techniques, such as decomposition, design of experiments, clusterization, and adaptive data sampling. More advanced approaches for ANN-based design exploiting microwave knowledge are briefly reviewed, including the hybrid EM-ANN approach, the prior knowledge input method, and knowledge-based neural networks. Computationally efficient neural SM methods for highly accurate EM-based design optimization are surveyed, contrasting different strategies for developing suitable (input and output) neural mappings. A high-level perspective is kept throughout the chapter, emphasizing the main ideas associated with these innovative techniques. A tutorial example using commercially available CAD tools is finally presented to illustrate the efficiency of the neural SM methods.

Keywords Computer-aided design (CAD) · Design automation · RF and microwave modeling · EM-based design optimization · Artificial neural network (ANN) · Space mapping (SM) · Knowledge-based neural network (KBNN) · Neural space mapping

1 Introduction

The modern era of artificial neural networks (ANNs) started in the 1940s and developed explosively in the 1980s, finding applications in many areas of science, en-

J.E. Rayas-Sánchez (✉)
Research Group on Computer-Aided Engineering of Circuits and Systems (CAECAS), Department of Electronics, Systems and Informatics, ITESO (Instituto Tecnológico y de Estudios Superiores de Occidente), Tlaquepaque, Jalisco, Mexico 45604
e-mail: erayas@iteso.mx

S. Koziel, L. Leifsson (eds.), *Surrogate-Based Modeling and Optimization*,
DOI 10.1007/978-1-4614-7551-4_7,

gineering, management, and other disciplines [1]. ANN applications in RF and microwave engineering have been reported since the 1990s [2]. Descriptions of ANNs and their key features, e.g., architectures, paradigms, training methods, data set formation, learning and generalization errors, and learning speed, in the context of RF and microwave CAD, have been extensively reported [3–8]. An excellent compilation and review of the main issues and initial applications of ANNs in the microwave arena was made by Burrascano and Mongiardo [8]. Patnaik and Mishra [9] developed an abbreviated review of ANN techniques to microwave modeling, design, and measurement problems (with some emphasis on antenna applications). Another excellent review on ANNs for RF circuits, high-speed interconnects, and microwave modeling is the work by Zhang et al. [10], which includes a comprehensive foundation on neural model development as well as a list of practical microwave neuromodels.

Neural networks have been widely used for modeling microwave devices and high-speed circuits in several innovative ways. The training and testing data for these models are typically obtained from full-wave electromagnetic (EM) simulators, from physics-based models, or from measurements. In the case of massive simulation tasks such as those required in RF/microwave nonlinear subsystems (e.g., front ends for mobile and personal communications), the training and testing data can be obtained from standard harmonic balance simulations using very detailed circuit models [11]. The resulting neural models are excellent vehicles for fast and accurate simulation.

By contrast, the use of neural networks for direct RF and microwave design by optimization is less developed. A review of the most relevant work in EM-based design and optimization of RF and microwave circuits exploiting ANNs can be found in [12], including measurement-based design of high-speed circuits using ANNs as well as synthesis neural networks, also called "inverse neural models."

This chapter focuses on the intersection of two major CAD technologies for modeling and design of RF and microwave circuits: ANNs and space mapping (SM), and it endeavors to address the most significant milestones at the intersection of these two consolidated technologies.

A brief introduction to neural networks is first presented, starting from elementary concepts associated to biological neurons up to the essential definitions concerning artificial neural models.

EM-based modeling and design optimization of microwave circuits using ANNs is subsequently addressed. The conventional and most widely used ANN approach for RF and microwave design optimization is explained, also referred to as the "black-box" approach, followed by indications of typical enhancing techniques, such as decomposition, design of experiments, clusterization, and adaptive data sampling.

More advanced approaches for ANN-based design exploiting microwave knowledge are briefly reviewed, including the hybrid EM-ANN approach, also called the difference method, the prior knowledge input (PKI) method, and knowledge-based neural networks (KBNNs).

Abbreviated descriptions of computationally efficient neural space mapping methods for highly accurate EM-based design optimization are formulated, contrasting different strategies for developing suitable neural mappings.

A high-level perspective is kept throughout the chapter, emphasizing the main ideas associated with these innovative techniques. A tutorial example using commercially available CAD tools is then presented to illustrate the efficiency of some of these neural space mapping methods.

2 Artificial Neural Networks

A biological brain can be seen as a highly complex, nonlinear, and massively parallel computer. The neuron is its fundamental processing unit; it receives and combines signals from many other neurons through filamentary paths called dendrites [13]. A simplified representation of a biological neuron is shown in Fig. 1.

Dendritic trees collect signals from many other neurons and transmit them to the main body of the neuronal cell, the soma, where most of the signal processing takes place. When the resultant processed signals reach a threshold, the neuron fires, transmitting an electrical signal, the action potential or induced local field [1], along a transmission channel called the axon (see Fig. 1). When the action potential reaches the axonic ending, chemical messengers known as neurotransmitters are released [13].

The connection between neurons is called a synapse. When a signal appears at the synapse, an electrical charge is generated, whose magnitude depends on the strength of all the incoming signals processed at the soma, which are weighted by factors that in general vary over time. Biological neurons also grow in size and connections over time through the natural learning process. Each neuron connects to many other neurons, forming complicated neural networks. A biological neural network may be conceived as a sophisticated signal processor, in which the strength of each connection or synapse (i.e., the synaptic weight) and the bias and threshold values of each neuron at steady state constitute the network's program.

Artificial neural networks (ANNs) emulate these biological processors: they are inspired by the ability of biological brains to learn from observation and generalize by abstraction.

2.1 Neuronal Models

Several basic models to approximate the behavior of a biological neuron in a simplified manner have been proposed [1]. The most widely used models for artificial neurons in RF and microwave engineering applications are the linear neuron, the inner product nonlinear neuron, and the Euclidean distance nonlinear neuron [14].

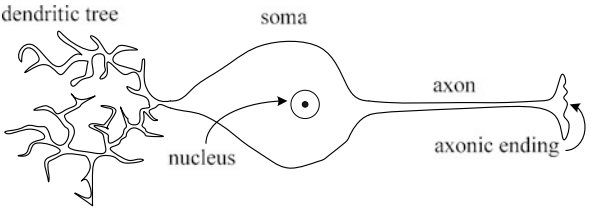


Fig. 1 Representation of a biological neuron

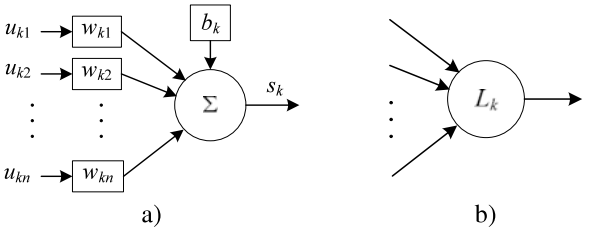


Fig. 2 Linear neuron: (**a**) model, (**b**) symbol

2.1.1 Linear Neuron

A general linear neuron can be modeled as shown in Fig. 2. Linear neurons are mostly used in the output layer of ANNs.

The k-th linear neuron (see Fig. 2) has n inputs and one output, where $\boldsymbol{u}_k = [u_{k1} \ \ldots \ u_{kn}]^T$ is the vector of inputs, which represent signals coming from other neurons, $\boldsymbol{w}_k = [w_{k1} \ \ldots \ w_{kn}]^T$ is the vector of weighting factors to represent the corresponding synapse strengths, and b_k is the bias or offset term. The output signal s_k is the activation potential given by

$$s_k = b_k + \boldsymbol{u}_k^T \boldsymbol{w}_k. \tag{1}$$

2.1.2 Inner-Product Nonlinear Neuron

The most popular nonlinear neuron, used mainly in the hidden layers of multiple-layer perceptrons (see Sect. 2.2), is the inner-product nonlinear neuron, whose model and symbol are shown in Fig. 3.

The inner-product nonlinear k-th neuron represented in Fig. 3 has n inputs and one output, where s_k is the activation potential or induced local field given by (1) and z_k is the output signal in neuron k calculated as

$$z_k = \varphi_k(s_k). \tag{2}$$

The purpose of the nonlinear function $\varphi_k(s_k)$, called the activation function or squashing function, is to ensure that the neuron's response is bounded, to emulate that the actual response of the biological neuron is conditioned or damped, as a result

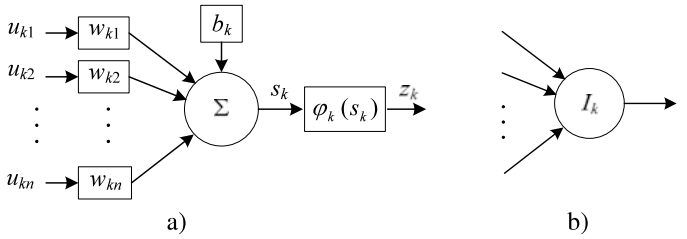


Fig. 3 Inner-product nonlinear neuron: (**a**) model, (**b**) symbol

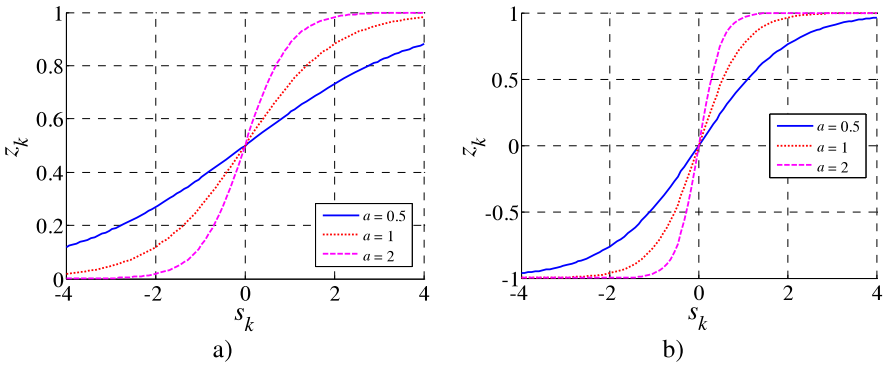


Fig. 4 Most popular activation functions: (**a**) sigmoid or logistic function (3), (**b**) hyperbolic tangent function (4)

of large or small stimuli. In the biological nervous system, conditioning of stimuli is continuously done by all sensory inputs [13]. Three popular nonlinearities for inner-product neurons are the sigmoid, the hard limiter, and the hyperbolic tangent [1].

If a sigmoid or logistic function is used, the response of neuron k is given by

$$z_k = \frac{1}{1 + e^{-as_k}}, \tag{3}$$

where the parameter a can be used to control the slope of the sigmoid, and therefore the amount of nonlinearity, as illustrated in Fig. 4a.

If a hyperbolic tangent is used, the neuronal response is given by

$$z_k = \frac{e^{as_k} - e^{-as_k}}{e^{as_k} + e^{-as_k}}. \tag{4}$$

The hyperbolic tangent activation function is illustrated in Fig. 4b, showing its behavior for different slope parameter values. The sigmoid and hyperbolic tangent functions are the most popular activation functions for ANNs because they are easy to implement and continuously differentiable.

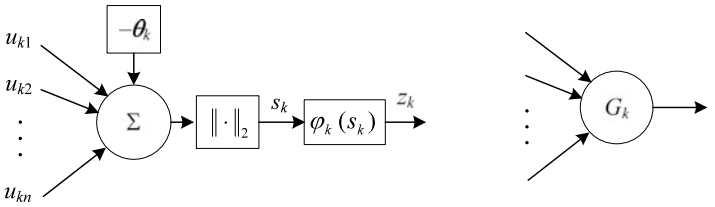


Fig. 5 Euclidean distance nonlinear neuron: (**a**) model, (**b**) symbol

2.1.3 Euclidean Distance Neuron

Euclidean distance neurons, shown in Fig. 5, are nonlinear neurons used to implement radial basis function (RBF) ANNs.

The Euclidean distance k-th neuron shown in Fig. 5 has n inputs and one output, where $\boldsymbol{\theta}_k = [\theta_{k1} \ \ldots \ \theta_{kn}]^T$ is the k-th center vector measured with respect to the Euclidean distance, s_k is the activation potential or induced local field given by

$$s_k = \|\boldsymbol{u}_k - \boldsymbol{\theta}_k\|_2, \tag{5}$$

and z_k is the output signal in neuron k calculated by (2).

The following nonlinear functions are of particular interest for RBF: multiquadratics, inverse multiquadratics, and Gaussian functions [1]. If a Gaussian function is used,

$$z_k = \varphi_k(s_k) = e^{-\frac{s_k^2}{2\sigma^2}} \tag{6}$$

for some $\sigma > 0$. Gaussian activation functions for several values of σ are illustrated in Fig. 6.

2.2 Three-Layer Perceptrons

Individual neurons can be connected in different ways, yielding different ANN topologies, also known as ANN paradigms. The most widely used paradigms are the multilayer feedforward perceptrons and the recurrent neural networks. Among the first kind, three-layer perceptrons (3LPs) are by far the most popular in RF and microwave applications [12].

The topology of a 3LP with n inputs, h hidden neurons, and m outputs is illustrated in Fig. 7. This ANN is denoted as a 3LP:n-h-m. The hidden layer is made up of nonlinear neurons, and the output layer typically uses linear neurons. The main input-output relationship is given by

$$\boldsymbol{v} = \boldsymbol{b}^{\mathrm{o}} + \boldsymbol{W}^{\mathrm{o}}\boldsymbol{\Phi}(\boldsymbol{s}), \quad \text{with } \boldsymbol{s} = \boldsymbol{b}^{\mathrm{h}} + \boldsymbol{W}^{\mathrm{h}}\boldsymbol{u}, \tag{7}$$

where matrix $\boldsymbol{W}^{\mathrm{o}} \in \Re^{m \times h}$ contains all the weighting factors of the output layer, $\boldsymbol{b}^{\mathrm{o}} \in \Re^{m}$ has the bias levels of the output neurons, $\boldsymbol{b}^{\mathrm{h}} \in \Re^{h}$ has the bias levels of the

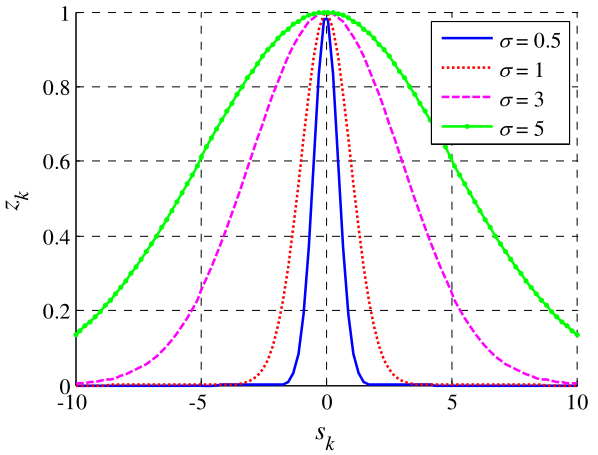


Fig. 6 Gaussian activation function typically used in radial basis function ANNs

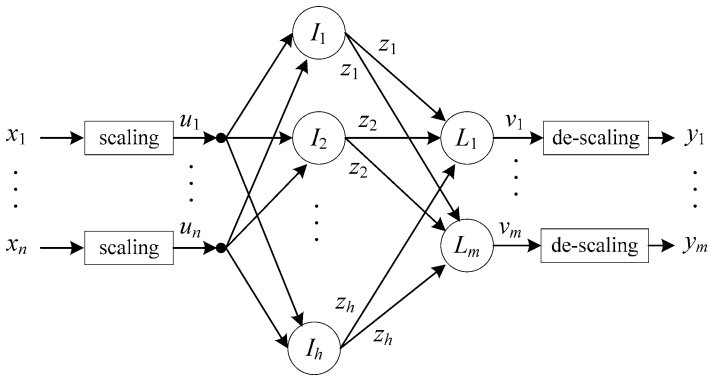


Fig. 7 Topology of a 3LP ANN. The hidden layer uses inner-product nonlinear neurons (see Fig. 3), while the output layer uses linear neurons (see Fig. 2)

hidden neurons, $\boldsymbol{W}^{\mathrm{h}} \in \Re^{n \times h}$ stores the weighting factors of the hidden layer, and the vector of activation functions $\boldsymbol{\Phi} \in \Re^{h}$ has the following structure:

$$\boldsymbol{\Phi}(\boldsymbol{s}) = \begin{bmatrix} \varphi(s_1) & \cdots & \varphi(s_h) \end{bmatrix}^T . \tag{8}$$

Figure 7 also indicates the scaling and descaling operations of the input and output signals. Scaling the input data allows controlling the relative importance of input parameters. Scaling the input/output training data also defines the dynamic range of the actual training region, and improves convergence during ANN training [2]. For instance, scaling training data between -1 and $+1$ requires

$$v_i = -1 + \frac{2(y_i - y_{i\,\min})}{(y_{i\,\max} - y_{i\,\min})}, \tag{9a}$$

$$u_i = -1 + \frac{2(x_i - x_{i\,\min})}{(x_{i\,\max} - x_{i\,\min})} \tag{9b}$$

for $i = 1, 2, \ldots, m$, while descaling the outputs requires

$$y_i = y_{i\,\min} + \frac{1}{2}(v_i + 1)(y_{i\,\max} - y_{i\,\min}). \tag{10}$$

2.3 Training ANNs

Learning, also called training, is the process by which an ANN adapts itself to different stimuli, by adjusting its synaptic weights and bias levels until it produces a desired response. Various learning strategies have been proposed to suit different ANN applications, including supervised learning, unsupervised learning, and competitive learning [1]. Sophisticated training strategies allow the possibility for the ANN to modify its own topology during learning, motivated by the fact that neurons in a biological brain can die and new synaptic connections can grow.

The problem of training an ANN can be formulated as an optimization problem, where ANN free parameters (weights, bias, etc.) are found such that the ANN responses match the desired responses in a region of interest. Some of the most used optimization techniques for ANN training are the delta rule algorithm, Boltzmann's algorithm, backpropagation, simulated annealing, and the Markovian technique [13]. In the RF and microwave engineering arena, ANNs are commonly trained by quasi-Newton methods, the scaled conjugate gradient method, and the Levenberg–Marquardt method.

3 The Conventional Approach to ANN-Based Design Optimization

The most common approach to design optimization of RF and microwave circuits using ANNs consists of developing a neuromodel of the original circuit within a certain region of the design parameters, and then applying conventional optimization techniques to the neuromodel to find the optimal solution that yields the desired response. This approach is illustrated in Fig. 8. Examples of this neural optimization approach can be found in [15–18].

The neuromodel is trained so that it approximates the fine model responses $\boldsymbol{R}_\mathrm{f}$ in a region of interest for the design parameters $\boldsymbol{x}_\mathrm{f}$ and operating conditions $\boldsymbol{\psi}$, as illustrated in Fig. 8a. The fine model responses $\boldsymbol{R}_\mathrm{f}$ are typically obtained from a full-wave EM simulator; in general, they represent the responses of an accurate but computationally expensive model (the term "fine model" comes from the space mapping literature [19–22]). The operating conditions are in vector $\boldsymbol{\psi}$, which can contain any combination of independent variables according to the nature of the simulation, such as the operating frequencies, bias levels, excitation levels, rise time,

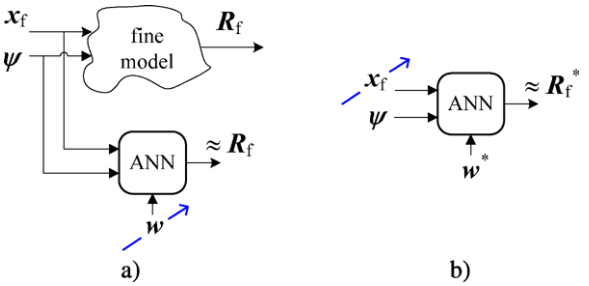


Fig. 8 Conventional ANN-based design optimization approach: (**a**) training the ANN to approximate the fine model responses in a region of interest, (**b**) design optimization with the available trained neuromodel

fall time, initial conditions, and temperature. Vector $\boldsymbol{w}$ contains the internal free parameters of the ANN (weighting factors, bias, etc.).

As mentioned before, the process of training the neuromodel (see Fig. 8a) can be formulated as an optimization problem, where the optimal vector of the ANN free parameters $\boldsymbol{w}^*$ is found by minimizing the difference between the ANN outputs and the fine model responses at all of the learning samples,

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \left\| \begin{bmatrix} \boldsymbol{e}_1^T & \dots & \boldsymbol{e}_L^T \end{bmatrix}^T \right\|, \tag{11}$$

$$\boldsymbol{e}_k(\boldsymbol{w}) = \boldsymbol{R}_f(\boldsymbol{x}_{fk}, \boldsymbol{\psi}) - \boldsymbol{N}(\boldsymbol{x}_{fk}, \boldsymbol{\psi}, \boldsymbol{w}), \quad k = 1, \dots, L, \tag{12}$$

where $\boldsymbol{N}$ represents the input–output relationship of the ANN (including scaling), $\boldsymbol{e}_k$ is the error vector at the k-th learning sample, L is the total number of learning base points, and $\| \cdot \|$ denotes a suitable norm (typically Euclidean, Manhattan, or Huber).

An adequate complexity of the ANN must be selected; in other words, the number of internal free parameters must be sufficiently large to achieve a small learning error, and small enough to avoid poor generalization performance. The latter occurs when too many free parameters in the ANN give large errors at points not seen during training. This generalization ability of the neuromodels is controlled during the training process by using separated testing data, also obtained from fine model evaluations.

Once the ANN is trained with sufficient learning samples and adequate generalization performance, i.e., once vector $\boldsymbol{w}^*$ is obtained from (11), the ANN can be used for fast and accurate simulations within the region of interest. It can also be used for inexpensive design optimization, to find an approximation of the optimal fine model solution $\boldsymbol{x}_f^*$ that yields the desired response $\boldsymbol{R}_f^* = \boldsymbol{R}_f(\boldsymbol{x}_f^*, \boldsymbol{\psi})$ (see Fig. 8b). This design problem consists in finding $\boldsymbol{x}_f^*$ such that

$$\boldsymbol{x}_f^* \approx \arg\min_{\boldsymbol{x}_f} U\left(\boldsymbol{N}\left(\boldsymbol{x}_f, \boldsymbol{\psi}, \boldsymbol{w}^*\right)\right), \tag{13}$$

where U is the objective function (typically minimax) expressed in terms of the design specifications. Problem (13) is usually solved by classical optimization methods, such as sequential quadratic programming (SQP).

The conventional approach to ANN-based design described before, which is also known as the "black-box" approach [23], has three important disadvantages: (1) the considerable time required to generate sufficient training and testing samples, (2) the unreliability of the optimal solution found in (13) when it lies outside the training region (due to the well-known poor extrapolation performance of ANNs), and (3) the "curse of dimensionality," which refers to the fact that the number of learning samples needed to approximate a function grows exponentially with the ratio between the dimensionality and its degree of smoothness [24]. This last disadvantage is very important when full-wave EM simulators or other computationally expensive models are employed for generating training data, since it implies that the number of EM model evaluations needed grows exponentially with the number of design parameters in the circuit.

A good alternative to reduce the size of the learning set in the black-box approach is to carefully select the learning points using the design of experiments (DOE) methodology, to ensure adequate parameter coverage, as in [25, 26]. Another way to speed up the learning process is by means of preliminary neural clusterization of similar responses using the self-organizing feature map (SOFM) approach, as in [8].

The conventional approach to ANN-based design optimization is very suitable when the device's physics is not fully understood and hence there is no empirical model available for the device, but the device's outputs for specified inputs are available, either from measurements or from accurate and reliable simulations. Another nice feature of the conventional neural optimization approach is its adequacy for full automation, to generate neuromodels for any desired level of accuracy within a user-defined region of interest, as in the work by Zhang et al. [27], where adaptive sampling schemes are incorporated to save fine model evaluations.

4 ANN-Based Design Optimization Exploiting Knowledge

The main disadvantages of the ANN-based conventional approach to design optimization can be alleviated by incorporating available knowledge into the neural network training scheme.

In most cases, the microwave knowledge incorporated into the neural network takes the form of a coarse model, usually obtained from an empirical, equivalent circuit model based on quasi-static approximations. It can also be incorporated by using analytical approximations or closed-form empirical functions. In any case, this knowledge representation is very fast to compute, but it is not accurate (the term "coarse model" comes from the space mapping literature [19–22]). Several innovative strategies to incorporate this knowledge are conceptually described in this section. A brief historical perspective of the evolution of these strategies can be found in [28].

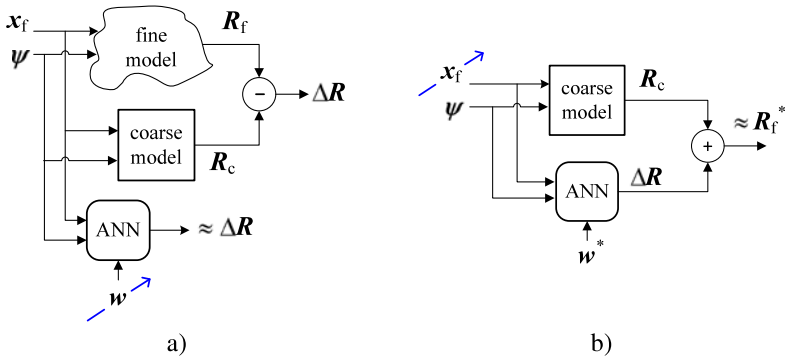


Fig. 9 The difference method for ANN-based design optimization approach: (**a**) training the ANN to approximate the difference between the fine and coarse model responses, (**b**) design optimization with the already-trained hybrid EM-ANN neuromodel

4.1 Hybrid EM-ANN Method

The hybrid EM-ANN method, also known as the difference method, trains the corresponding neural network to approximate the difference in the responses between an available coarse model and the fine model, as illustrated in Fig. 9a. Once the ANN model is trained, a shunt arrangement between the coarse model and the ANN is employed for fast design optimization (see Fig. 9b).

Training the neuromodel in the hybrid EM-ANN method can be formulated as (11) with the error functions calculated by

$$\boldsymbol{e}_k(\boldsymbol{w}) = \left[\boldsymbol{R}_\mathrm{f}(\boldsymbol{x}_{\mathrm{f}k}, \boldsymbol{\psi}) - \boldsymbol{R}_\mathrm{c}(\boldsymbol{x}_{\mathrm{f}k}, \boldsymbol{\psi})\right] - \boldsymbol{N}(\boldsymbol{x}_{\mathrm{f}k}, \boldsymbol{\psi}, \boldsymbol{w}). \tag{14}$$

The design optimization problem is then formulated as

$$\boldsymbol{x}_\mathrm{f}^* \approx \arg\min_{\boldsymbol{x}_\mathrm{f}} U\left(\boldsymbol{R}_\mathrm{c}(\boldsymbol{x}_\mathrm{f}, \boldsymbol{\psi}) + \boldsymbol{N}\left(\boldsymbol{x}_\mathrm{f}, \boldsymbol{\psi}, \boldsymbol{w}^*\right)\right). \tag{15}$$

Application examples of this hybrid EM-ANN approach are given in [29, 30], including the design of a two-layer end-coupled bandpass filter. In some cases, the number of fine model simulations needed to train the ANN can be significantly reduced by this approach. However, the reduction in training samples is achieved only when the mapping from the difference between the fine and coarse model responses to the input parameters is simpler than the original target relationship [31], which is not always the case, as demonstrated in [12].

4.2 Prior Knowledge Input Method

In the prior knowledge input (PKI) method the coarse model responses are used as inputs for the ANN in addition to the design parameters and independent variables, as shown in Fig. 10. The neural network is trained such that its response is

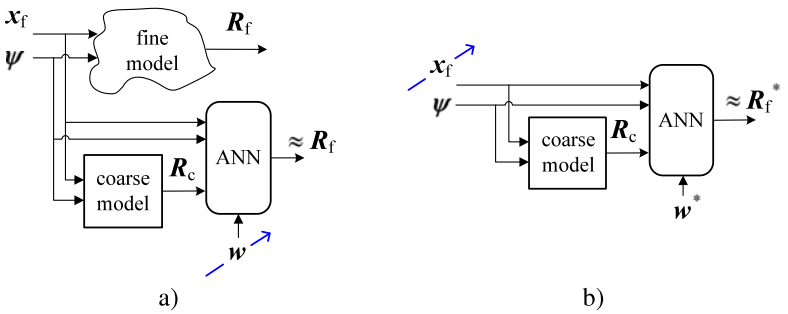


Fig. 10 The PKI method for design optimization: (**a**) training the ANN to approximate the fine model responses, considering the coarse model responses as additional inputs to the ANN; (**b**) design optimization with the already-trained PKI neuromodel

as close as possible to the fine model response for all the data in the training set (see Fig. 10a). Once it is trained, it can be combined with the original coarse model for inexpensive optimization (see Fig. 10b). It has been reported [31, 32] that the PKI approach exhibits better accuracy than the hybrid EM-ANN approach, at the expense of a more complex ANN. The PKI method is used in [33] to optimize a coplanar waveguide (CPW) patch/slot antenna on Duroid substrate.

4.3 Knowledge-Based ANN Approach

Developed by Zhang et al. [34], the knowledge-based neural network (KBNN) approach modifies the internal structure of the ANN to incorporate available knowledge, as illustrated in Fig. 11. This knowledge usually takes the form of microwave empirical or analytical equations.

Knowledge-based ANNs have non-fully connected architectures, with one or several layers assigned to the microwave knowledge in the form of single or multidimensional vector functions, usually obtained from available closed-form expressions based on quasi-static approximations.

By inserting the microwave circuit empirical formulas into the neural network structure, the empirical formulas can also be adjusted as part of the overall neural network training process. Since these empirical functions are used for some neurons instead of standard activation functions, KBNNs do not follow a typical multilayer perceptron architecture and are trained using methods other than the conventional backpropagation [34]. In Fig. 11a, vector $\boldsymbol{w}$ contains not only the typical free parameters of an ANN (weights, bias, etc.), but also the adjustable parameters of the microwave empirical functions.

KBNNs have been extensively used for developing models of microwave circuits [2, 34]. Once a KBNN model is appropriately trained, it can be used as an accurate and inexpensive model for realizing conventional design optimization (see Fig. 11b).

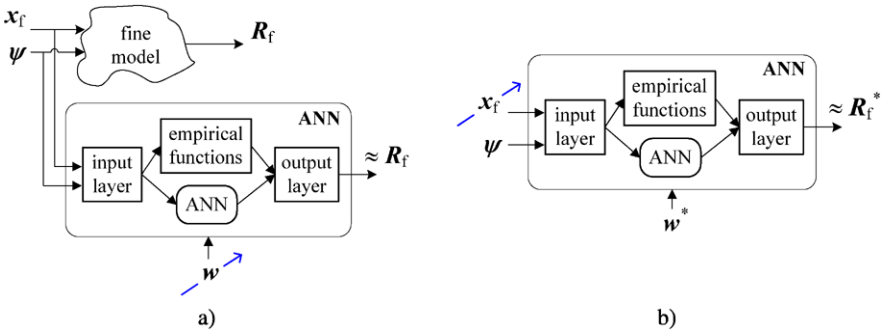


Fig. 11 The KBNN approach to design optimization: (**a**) training the KBNN model (the empirical functions and formulas are embedded in the ANN internal structure), (**b**) designing with the already-trained KBNN model

The KBNN method is perhaps the most matured and automated technique for developing neuromodels of microwave circuits. The automatic model generation (AMG) scheme [35] allows the development of knowledge-based neuromodels with intelligent data sampling, adaptive ANN sizing, and under-learning and over-learning detection.

4.4 Neural Space Mapping Methods

In general, neural space mapping methods can be regarded as a special case of ANN-based modeling and design exploiting knowledge [12, 36]. Although KBNN methods have also incorporated space mapping techniques, as in [35], they differ with respect to pure neural space mapping methods by the fact that the latter do not modify the internal architecture of the ANN [28]. In neural space mapping methods the ANN is a classical feedforward perceptron with conventional nonlinear activation functions, and it is used to implement a mapping function, either at the level of the design parameters (input space mapping) or at the level of the model responses (output space mapping).

The initial work for efficient EM-based modeling exploiting space mapping and neural networks originated with Bandler et al. in [37]. Similarly to KBNN, neural space mapping methods decrease the cost of training, improve the generalization ability, and reduce the complexity of the ANN with respect to conventional neuromodeling approaches.

The essential idea of neural space mapping methods is to implement with an ANN the mapping between fine and coarse models, exploiting the well-known ability of ANNs to model high-dimensional and highly nonlinear problems, obtaining enhanced coarse models valid over large regions of design parameters. This mapping can be developed at the input (see Fig. 12) or at the output (see Fig. 13).

In the output neural space mapping approach an input mapping function $\boldsymbol{P}$ is still used, as illustrated in Fig. 13. However, in this case the function $\boldsymbol{P}$ is usually linear,

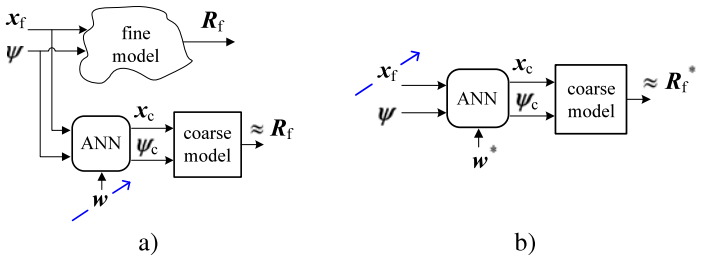


Fig. 12 Neural input space mapping: (**a**) training the ANN mapping function, (**b**) designing with the already-trained neural space-mapped model

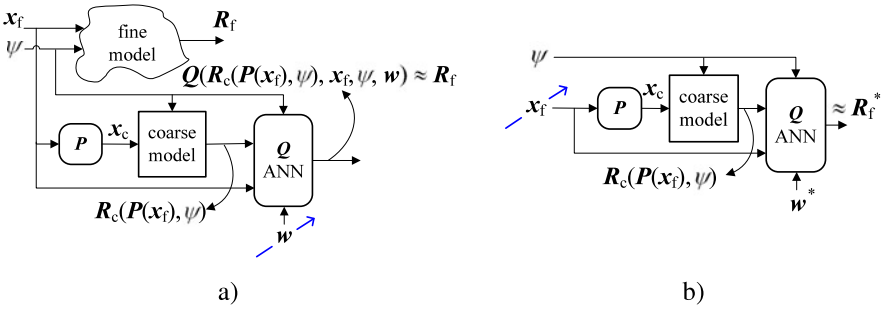


Fig. 13 Neural output space mapping: (**a**) training the ANN mapping function, (**b**) designing with the already-trained neural space-mapped model

since the output neural network $\boldsymbol{Q}$ is able to eliminate any residual between the coarse and fine model responses [38], which is more useful for statistical analysis and yield estimation. In a strict sense, $\boldsymbol{P}$ is not required; however, it is included to decrease the level of complexity in $\boldsymbol{Q}$, improving the generalization performance of the output neural network. Application examples of this approach can be found in [39].

5 Design Optimization by Neural Space Mapping: Example

An example of practical design optimization using neural input space mapping is now described. First, the general formulation is reviewed. Next, the structure example is described, along with its coarse and fine model representations using commercially available simulators. Finally, the results are provided.

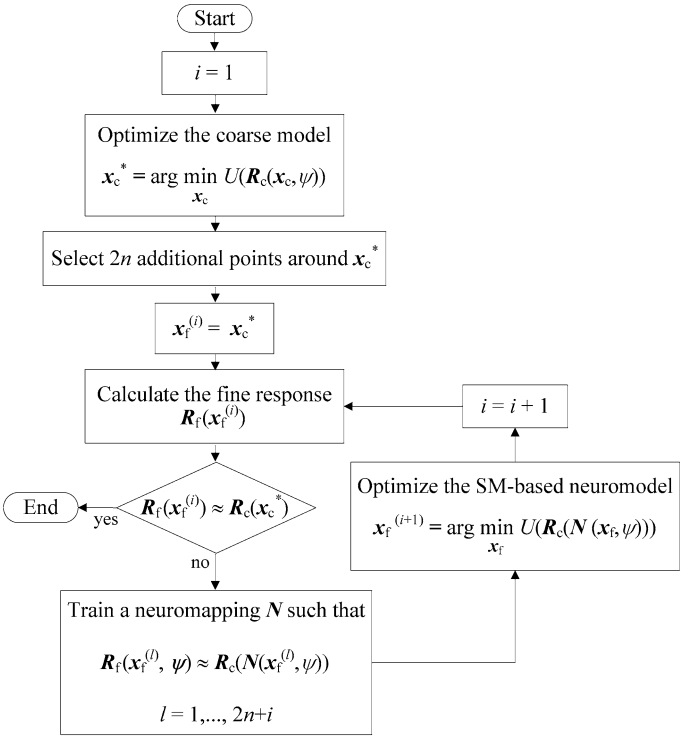


Fig. 14 Flow diagram for neural input space mapping design optimization (see Fig. 12)

5.1 Neural Input Space Mapping Algorithm

The general concept of neural input space mapping is illustrated in Fig. 12, where the mapping function from the fine to the coarse model design parameter space is implemented by an ANN. A simplified flow diagram for neural input space mapping optimization is illustrated in Fig. 14. This technique was the first algorithmic formulation of ANN-based design of microwave circuits [40].

The algorithm starts by applying conventional optimization to the coarse model to find the optimal coarse model solution $\boldsymbol{x}_c^*$ that yields a target response $\boldsymbol{R}^* = \boldsymbol{R}_c(\boldsymbol{x}_c^*)$. In Fig. 14, $U(\cdot)$ represents the same objective function employed in (13) and (15). The initial training set to develop a space mapping (SM)-based neuromodel [37] is defined by taking $2n$ additional base points in a star distribution [41] centered at $\boldsymbol{x}_c^*$, where n is the number of design parameters ($\boldsymbol{x}_c, \boldsymbol{x}_f \in \Re^n$). Training the neuromapping is formulated as in (11), but we replace (12) by the error vector function

$$\boldsymbol{e}_k(\boldsymbol{w}) = \boldsymbol{R}_f(\boldsymbol{x}_{fk}, \boldsymbol{\psi}) - \boldsymbol{R}_c\big(\boldsymbol{N}(\boldsymbol{x}_{fk}, \boldsymbol{\psi}, \boldsymbol{w})\big), \quad k = 1, \ldots, (2n + i). \tag{16}$$

Once an SM-based neuromodel is trained (see Fig. 12a), it is used as an improved coarse model, optimizing its design parameters to generate the desired response. The solution to the optimization problem

$$\boldsymbol{x}_{\mathrm{f}}^{(i+1)} = \arg\min_{\boldsymbol{x}_{\mathrm{f}}} U\left(\boldsymbol{R}_{\mathrm{c}}\left(\boldsymbol{N}\left(\boldsymbol{x}_{\mathrm{f}}, \boldsymbol{\psi}, \boldsymbol{w}^{*}\right)\right)\right) \tag{17}$$

becomes the next iterate and is included in the learning set, as indicated in Fig. 14.

The fine model response at the new point is calculated and compared with the desired response. If they are not close enough, the SM-based neuromodel is re-trained over the extended set of learning samples and the algorithm continues; otherwise, the algorithm terminates.

An interesting feature of this algorithm is that the independent variable $\boldsymbol{\psi}$ can also be transformed through the neural network in order to improve the alignment between the fine and coarse model responses, as shown in Fig. 12. This allows us, for instance, to define a frequency-sensitive mapping. Additionally, the mapping $\boldsymbol{N}$ can be defined to map only some of the design parameters. This flexibility yields a number of different techniques to establish the neuromapping $\boldsymbol{N}$ [37]. The neural space mapping approach has been successfully adapted for nonlinear modeling [42, 43], and extended to large-signal statistical modeling of nonlinear microwave devices [44].

When training the neuromapping $\boldsymbol{N}$ in the flow diagram of Fig. 14, the complexity of the neural network $\boldsymbol{N}$ is gradually increased, starting with a three-layer perceptron (3LP) with zero hidden neurons (linear mapping). The number of hidden neurons h is increased according to the total learning error at the i-th iteration. This is realized to control the generalization performance of the current neural network $\boldsymbol{N}$. A better strategy to do this consists of using two-layer perceptrons (2LPs) with optimized nonlinearity, as in [45]. A comparison between these two strategies to control the generalization performance of $\boldsymbol{N}$ is realized in the following example.

5.2 Microstrip Notch Filter with Mitered Bends and Open Stubs

A microstrip notch filter with mitered bends and open stubs is illustrated in Fig. 15 [39]. L_{o} is the open stubs length, L_{c} is the length of the coupled lines, and S_{g} is the separation gap. The width W_{50} is the same for all the sections as well as for the input and output lines, of length L_{p}. A substrate with thickness H and relative dielectric constant ε_{r} is used.

The design parameters are $\boldsymbol{x}_{\mathrm{f}} = [L_{\mathrm{c}}\ L_{\mathrm{o}}\ S_{\mathrm{g}}]^{\mathrm{T}}$ (mil). The preassigned parameter values are $H = 10$ mil, $W_{50} = 31$ mil, and $\varepsilon_{\mathrm{r}} = 2.2$ (RT Duroid 5880, with loss tangent $\tan\delta = 0.0009$). Lossless metals are considered.

The design specifications are $|S_{21}| \leq 0.05$ for frequencies between 13.19 GHz and 13.21 GHz, and $|S_{21}| \geq 0.95$ for frequencies below 13 GHz and above 13.4 GHz.

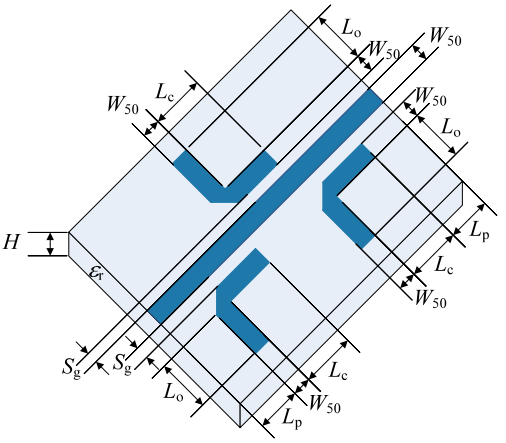


Fig. 15 Microstrip notch filter with mitered bends and open stubs

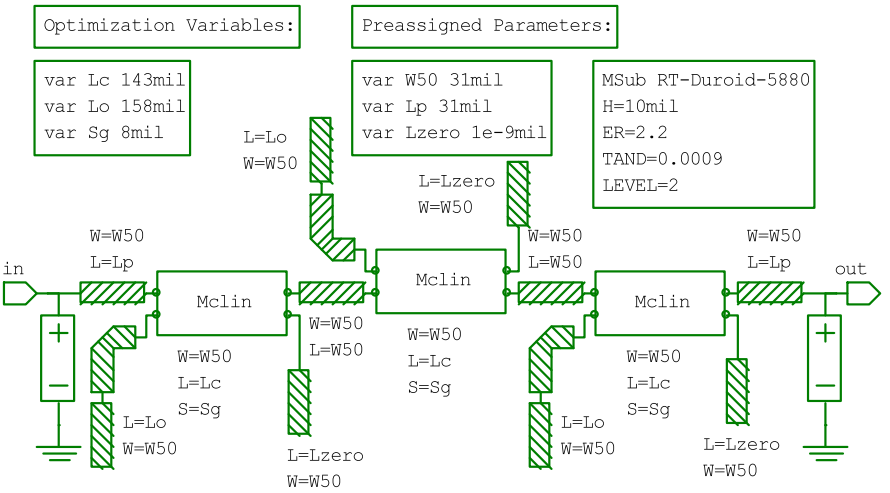


Fig. 16 Coarse model of the microstrip notch filter with mitered bends and open stubs, as implemented in APLAC

The coarse model is illustrated in Fig. 16. It consists of an equivalent distributed circuit implemented in APLAC,[1] using its built-in microstrip circuit models available for lines, open circuited lines, coupled lines, and mitered bends.

[1] APLAC version 8.10 2005, APLAC Solutions Corporation, now AWR Corporation, 1960 E. Grand Avenue, Suite 430, El Segundo, CA 90245.

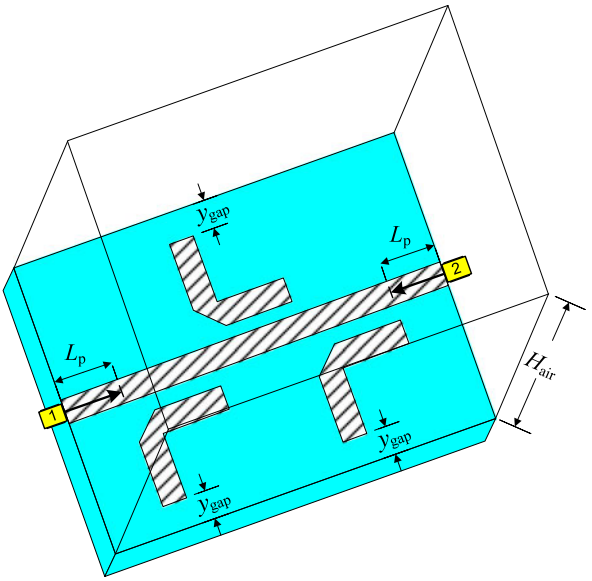


Fig. 17 Fine model of the microstrip notch filter with mitered bends and open stubs, as implemented in Sonnet

The fine model is implemented in the full-wave EM simulator Sonnet,[2] whose simulation box is illustrated in Fig. 17. It uses a very high resolution grid (cell size of 0.5 mil × 0.5 mil) with $H_{air} = 60$ mil, $L_p = 1/2(L_o + L_c)$, and $y_{gap} = L_o$. By using these values of y_{gap} and H_{air}, the effects of unwanted EM interaction and potential resonances with Sonnet's box are disabled.

After optimizing the coarse model for this example using the sequential quadratic programming (SQP) method available in MATLAB,[3] the optimal coarse model solution is $\boldsymbol{x}_c^* = [143\ 158\ 8]^T$ (mil). To generate the learning base points, 6 in this case, a 3 % of deviation for L_c and L_o is used, while a 50 % for S_g is applied, with respect to their values at $\boldsymbol{x}_c^*$. The coarse and fine model responses at $\boldsymbol{x}_c^*$ are shown in Fig. 18.

To contrast both strategies for controlling the generalization performance of the neuromapping, as mentioned in the previous section, this example is solved using 3LPs with gradually increasing complexity [40], as well as 2LPs with optimized nonlinearity [45]. In the latter case, after only one additional evaluation of the fine model, the SM solution $\boldsymbol{x}_f^{SM} = [143\ 159.5\ 6.5]^T$ (mil) is found, while the former approach requires 3 additional fine model simulations, as seen in Fig. 19, where the evolution of the fine model objective function values for each strategy is shown.

[2] ***em***™ Suite version 12.52 2009, Sonnet Software, Inc., 100 Elwood Davis Road, North Syracuse, NY 13212.

[3] MATLAB Optimization Tolbox, Version 7.4.1 (R2007a), The Mathworks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098.

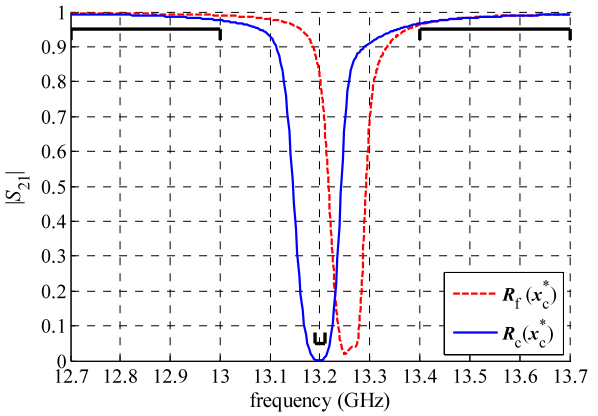


Fig. 18 Coarse and fine model responses at the optimal coarse model design of the microstrip notch filter

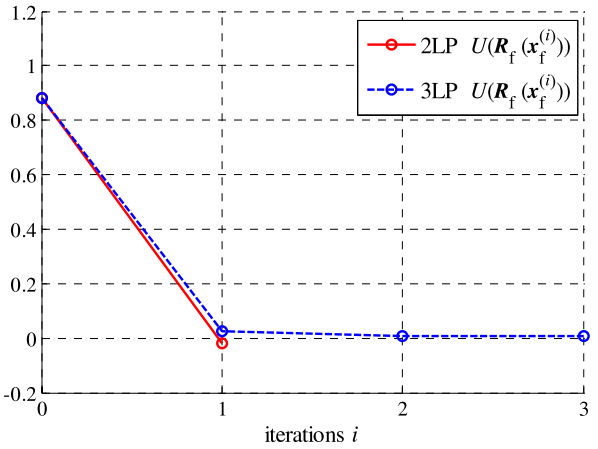


Fig. 19 Fine model objective function values at each neural input space mapping iteration for the microstrip notch filter

Table 1 summarizes the fine model design parameter values at each iteration of the algorithms, as well as the corresponding fine model objective function values. It is seen that formulation using 2LPs with optimized nonlinearity yields a better final objective function value in fewer fine model evaluations.

Finally, the target response and fine model responses evaluated at the space mapping solutions are shown in Fig. 20. It is seen that the space-mapped response found by using 2LPs with optimized nonlinearity is significantly better than the one found by using 3LPs with a gradually increasing number of hidden neurons.

Table 1 Design parameters and fine model objective function values at each iteration of neural input space mapping for the microstrip notch filter

Iteration	Using 3LP		Using 2LP	
	$\boldsymbol{x}_{\mathrm{f}}^{(i)}$	$U(i)$	$\boldsymbol{x}_{\mathrm{f}}^{(i)}$	$U(i)$
0	$[143\ 158\ 8]^{\mathrm{T}}$	0.8807	$[143\ 158\ 8]^{\mathrm{T}}$	0.8807
1	$[142.5\ 160\ 9.5]^{\mathrm{T}}$	0.02354	$[143\ 159.5\ 6.5]^{\mathrm{T}}$	−0.0215
2	$[143.5\ 159\ 9]^{\mathrm{T}}$	0.00580		
3	$[143.5\ 159\ 9]^{\mathrm{T}}$	0.00580		

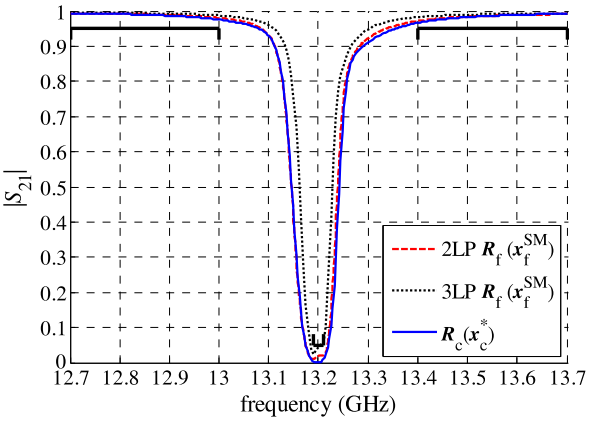


Fig. 20 Final results after applying neural input space mapping optimization to the microstrip notch filter, using 3LPs with gradually increasing complexity and 2LPs with optimized nonlinearity

6 Conclusion

A tutorial review of artificial neural networks and space mapping for EM-based modeling and design optimization of RF and microwave circuits has been presented in this chapter. The emphasis was on the most significant milestones at the intersection of these two consolidated CAD technologies. A brief introduction to the essentials of artificial neural networks was presented. The conventional or "black-box" approach for RF and microwave design optimization was explained, indicating typical enhancing techniques. More advanced methods for ANN-based design exploiting microwave knowledge were reviewed. Neural space mapping methods for highly accurate EM-based design optimization were surveyed, contrasting several different strategies, and the main ideas associated with these advanced techniques were highlighted. A tutorial example was presented to illustrate some of the most advanced methods for design optimization using neural space mapping techniques.

References

1. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice-Hall, Englewood Cliffs (1999)
2. Zhang, Q.J., Gupta, K.C.: Neural Networks for RF and Microwave Design. Artech House, Norwood (2000)
3. Zaabab, A.H., Zhang, Q.J., Nakhla, M.S.: Device and circuit-level modeling using neural networks with faster training based on network sparsity. IEEE Trans. Microw. Theory Tech. **45**, 1696–1704 (1997)
4. Devabhaktuni, V.K., Xi, C., Wang, F., Zhang, Q.J.: Robust training of microwave neural models. In: IEEE MTT-S Int. Microwave Symp. Dig., Anaheim, CA, June 1999, pp. 145–148 (1999)
5. Zhang, Q.J., Devabhaktuni, V.K.: Neural network structures for EM/microwave modeling. In: IEEE AP-S Int. Symp. Dig, Orlando, FL, July, pp. 2576–2579 (1999)
6. Wang, F., Devabhaktuni, V.K., Xi, C., Zhang, Q.J.: Neural network structures and training for RF and microwave applications. Int. J. RF Microw. Comput.-Aided Eng. **11**, 216–240 (1999)
7. Vai, M.M., Prasad, S.: Applications of neural networks optimized by the genetic algorithm to microwave systems. In: IEEE AP-S Int. Symp. Dig., Orlando, FL, July, pp. 2580–2583 (1999)
8. Burrascano, P., Mongiardo, M.: A review of artificial neural networks applications in microwave CAD. Int. J. RF Microw. Comput.-Aided Eng. **9**, 158–174 (1999)
9. Patnaik, A., Mishra, R.K.: ANN techniques in microwave engineering. IEEE Microw. Mag. **1**, 55–60 (2000)
10. Devabhaktuni, V.K., Yagoub, M.C.E., Fang, Y., Xu, J.J., Zhang, Q.J.: Neural networks for microwave modeling: model development issues and nonlinear modeling techniques. Int. J. RF Microw. Comput.-Aided Eng. **11**, 4–21 (2001)
11. Rizzoli, V., Neri, A., Masotti, D., Lipparini, A.: A new family of neural network-based bidirectional and dispersive behavioral models for nonlinear RF/microwave subsystems. Int. J. RF Microw. Comput.-Aided Eng. **12**, 51–70 (2002)
12. Rayas-Sánchez, J.E.: EM-based optimization of microwave circuits using artificial neural networks: the state of the art. IEEE Trans. Microw. Theory Tech. **52**, 420–435 (2004)
13. Kartalopoulos, S.V.: Understanding Neural Networks and Fuzzy Logic. IEEE Press, New York (1996)
14. Zhang, Q.J., Rayas-Sánchez, J.E.: Fast parametric models for EM design using neural networks and space mapping. In: IEEE MTT-S Int. Microwave Symp. Workshop Notes and Short Courses, Atlanta, GA, June (2008)
15. Zaabab, A.H., Zhang, Q.J., Nakhla, M.S.: A neural network modeling approach to circuit optimization and statistical design. IEEE Trans. Microw. Theory Tech. **43**, 1349–1358 (1995)
16. Veluswami, A., Nakhla, M.S., Zhang, Q.J.: The application of neural networks to EM-based simulation and optimization of interconnects in high-speed VLSI circuits. IEEE Trans. Microw. Theory Tech. **45**, 712–723 (1997)
17. Burrascano, P., Dionigi, M., Fancelli, C., Mongiardo, M.: A neural network model for CAD and optimization of microwave filters. In: IEEE MTT-S Int. Microwave Symp. Dig., Baltimore, MD, June, pp. 13–16 (1998)
18. Mishra, R.K., Patniak, A.: Neural network-based CAD model for the design of square-patch antennas. IEEE Trans. Antennas Propag. **46**, 1890–1891 (1998)
19. Bandler, J.W., Biernacki, R.M., Chen, S.H., Grobelny, P.A., Hemmers, R.H.: Space mapping technique for electromagnetic optimization. IEEE Trans. Microw. Theory Tech. **42**, 2536–2544 (1994)
20. Bandler, J.W., Biernacki, R.M., Chen, S.H., Hemmers, R.H., Madsen, K.: Electromagnetic optimization exploiting aggressive space mapping. IEEE Trans. Microw. Theory Tech. **41**, 2874–2882 (1995)
21. Bandler, J.W., Cheng, Q., Dakroury, S.A., Mohamed, A.S., Bakr, M.H., Madsen, K., Søndergaard, J.: Space mapping: the state of the art. IEEE Trans. Microw. Theory Tech. **52**, 337–361 (2004)

22. Koziel, S., Cheng, Q.S., Bandler, J.W.: Space mapping. IEEE Microw. Mag. **9**, 105–122 (2008)
23. Vai, M., Prasad, S.: Neural networks in microwave circuit design—beyond black-box models. Int. J. RF Microw. Comput.-Aided Eng. **9**, 187–197 (1999)
24. Stone, C.J.: Optimal global rates of convergence for nonparametric regression. Ann. Stat. **10**, 1040–1053 (1982)
25. Creech, G.L., Paul, B.J., Lesniak, C.D., Jenkins, T.J., Calcatera, M.C.: Artificial neural networks for fast and accurate EM-CAD of microwave circuits. IEEE Trans. Microw. Theory Tech. **45**, 794–802 (1997)
26. Joodaki, M., Kompa, G.: A systematic approach to a reliable neural model for pHEMT using different numbers of training data. In: IEEE MTT-S Int. Microwave Symp. Dig., Seattle, WA, June, pp. 1105–1108 (2002)
27. Devabhaktuni, V., Yagoub, M.C.E., Zhang, Q.J.: A robust algorithm for automatic development of neural-network models for microwave applications. IEEE Trans. Microw. Theory Tech. **49**, 2282–2291 (2001)
28. Rayas-Sánchez, J.E., Zhang, Q.J.: On knowledge-based neural networks and neuro-space mapping. In: IEEE MTT-S Int. Microwave Symp. Dig, Montreal, Canada, June, pp. 1–3 (2012)
29. Watson, P.M., Gupta, K.C.: EM-ANN models for microstrip vias and interconnects in dataset circuits. IEEE Trans. Microw. Theory Tech. **44**, 2495–2503 (1996)
30. Cho, C., Gupta, K.C.: EM-ANN modeling of overlapping open-ends in multilayer lines for design of bandpass filters. In: IEEE AP-S Int. Symp. Dig., Orlando, FL, July, pp. 2592–2595 (1999)
31. Watson, P.M., Gupta, K.C., Mahajan, R.L.: Applications of knowledge-based artificial neural network modeling to microwave components. Int. J. RF Microw. Comput.-Aided Eng. **9**, 254–260 (1999)
32. Watson, P.M., Gupta, K.C., Mahajan, R.L.: Development of knowledge based artificial neural networks models for microwave components. In: IEEE MTT-S Int. Microwave Symp. Dig., Baltimore, MD, June, pp. 9–12 (1998)
33. Watson, P.M., Creech, G.L., Gupta, K.C.: Knowledge based EM-ANN models for the design of wide bandwidth CPW patch/slot antennas. In: IEEE AP-S Int. Symp. Dig., Orlando, FL, July, pp. 2588–2591 (1999)
34. Wang, F., Zhang, Q.J.: Knowledge based neuromodels for microwave design. IEEE Trans. Microw. Theory Tech. **45**, 2333–2343 (1997)
35. Devabhaktuni, V.K., Chattaraj, B., Yagoub, M.C.E., Zhang, Q.J.: Advanced microwave modeling framework exploiting automatic model generation, knowledge neural networks, and space mapping. IEEE Trans. Microw. Theory Tech. **51**, 1822–1833 (2003)
36. Rayas-Sánchez, J.E.: Neural space mapping methods for modeling and design of microwave circuits. Ph.D. Thesis, McMaster University, Hamilton, Canada L8S 4K1 (2001). www.sos.mcmaster.ca/theses.htm
37. Bandler, J.W., Ismail, M.A., Rayas-Sánchez, J.E., Zhang, Q.J.: Neuromodeling of microwave circuits exploiting space mapping technology. IEEE Trans. Microw. Theory Tech. **47**, 2417–2427 (1999)
38. Rayas-Sánchez, J.E., Gutiérrez-Ayala, V.: EM-based statistical analysis and yield estimation using linear-input and neural-output space mapping. In: IEEE MTT-S Int. Microwave Symp. Dig., San Francisco, CA, June, pp. 1597–1600 (2006)
39. Rayas-Sánchez, J.E., Gutiérrez-Ayala, V.: EM-based Monte Carlo analysis and yield prediction of microwave circuits using linear-input neural-output space mapping. IEEE Trans. Microw. Theory Tech. **54**, 4528–4537 (2006)
40. Bakr, M.H., Bandler, J.W., Ismail, M.A., Rayas-Sánchez, J.E., Zhang, Q.J.: Neural space mapping optimization of EM microwave structures. In: IEEE MTT-S Int. Microwave Symp. Dig., Boston, MA, June, pp. 879–882 (2000)
41. Biernacki, R.M., Bandler, J.W., Song, J., Zhang, Q.J.: Efficient quadratic approximation for statistical design. IEEE Trans. Circuits Syst. **36**, 1449–1454 (1989)

42. Zhang, L., Xu, J.J., Yagoub, M.C.E., Ding, R., Zhang, Q.J.: Neuro-space mapping technique for nonlinear device modeling and large signal simulation. In: IEEE MTT-S Int. Microwave Symp. Dig., Philadelphia, PA, June, pp. 173–176 (2003)
43. Zhang, L., Xu, J.J., Yagoub, M.C.E., Ding, R.T., Zhang, Q.J.: Efficient analytical formulation and sensitivity analysis of neuro-space mapping for nonlinear microwave device modeling. IEEE Trans. Microw. Theory Tech. **53**, 2752–2767 (2005)
44. Zhang, L., Zhang, Q.J., Wood, J.: Statistical neuro-space mapping technique for large-signal modeling of nonlinear devices. IEEE Trans. Microw. Theory Tech. **56**, 2453–2467 (2008)
45. Gutiérrez-Ayala, V., Rayas-Sánchez, J.E.: Neural input space mapping optimization based on nonlinear two-layer perceptrons with optimized nonlinearity. Int. J. RF Microw. Comput.-Aided Eng. **20**, 512–526 (2010)

Model-Based Variation-Aware Integrated Circuit Design

Ting Zhu, Mustafa Berke Yelten, Michael B. Steer, and Paul D. Franzon

Abstract Modern integrated circuit designers must deal with complex design and simulation problems while coping with large device to device parametric variations and often imperfect information. This chapter presents surrogate model-based methods to generate circuit performance models for design, device models, and high-speed input-output (IO) buffer macromodels. Circuit performance models are built with design parameters and parametric variations, and they can be used for fast and systematic design space exploration and yield analysis. Surrogate models of the main device characteristics are generated in order to assess the effects of variability in analog circuits. The variation-aware IO buffer macromodel integrates surrogate modeling and a physically based model structure. The new IO macromodel provides both good accuracy and scalability for signal integrity analysis.

Keywords Surrogate modeling · Design exploration · Integrated circuit · Model-based design · Performance model · IO model · Device model · Variation aware · Circuit optimization · Design aids · Yield analysis · Self-calibrated circuit

1 Introduction

As integrated circuit (IC) technology continues to scale down, process variations become increasingly critical and lead to large variances of important transistor parameters. These large process uncertainties have caused significant yield loss. In addition, environmental variations (such as changes in supply voltage and temperature) and reliability issues contribute to further yield reduction and make it more challenging to create a reliable, robust design. In coping with these problems in circuit design, it is important to consider the effects of variations in circuit modeling and design analysis at an early stage. However this is a nontrivial task. In this chapter, surrogate modeling is applied to handle the complexities in variation-aware circuit macromodeling, model-based design optimization, high-speed IO macromodeling

T. Zhu (✉) · M.B. Yelten · M.B. Steer · P.D. Franzon
Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, USA
e-mail: tzhu@ncsu.edu

S. Koziel, L. Leifsson (eds.), *Surrogate-Based Modeling and Optimization*,
DOI 10.1007/978-1-4614-7551-4_8,

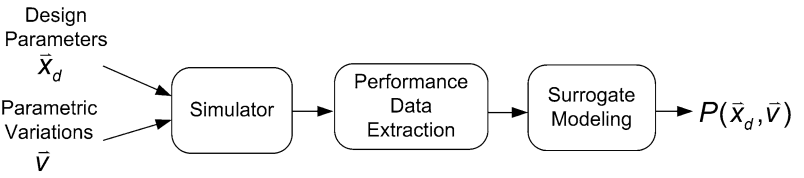


Fig. 1 Circuit performance modeling

and device modeling. This chapter presents the advantages of using surrogate modeling in enhancing the accuracy, flexibility, and efficiency in those applications.

2 Circuit Performance Modeling and Design Optimization

2.1 Overview

Today circuit designers are confronted with large design spaces and many design variables whose relationships must be analyzed. In this situation, tasks such as sensitivity analysis, design space exploration, and visualization become difficult, even if a single simulation takes only a short period of time. The analyses are becoming impractical, as some of the circuit simulations are computationally expensive and time-consuming. In addition a designer must not only search for the optimal design with the nominal conditions, but must also carefully consider the circuit robustness in the presence of variations. However, the fulfillment of all these requirements introduces more complications in circuit design. One way to reduce design complexities and costs is to build performance models which can be used as replacements for the real circuit performance responses.

In the approach described here, performance models are built by directly approximating circuit performance parameters (e.g., S-parameter, gain, power consumption, and noise figure) with design variables (e.g., transistor size, bias voltage, and current) and parametric variations (e.g., V_{th}, t_{ox}, L_{eff}) and then these models are used to drive the design. The modeling concept is illustrated in Fig. 1. This method is data-driven and black-box by nature, and thus it can be applied to a wide range of circuit design problems. When performance models are available, it is possible to explore the design space with these cheap-to-run performance models. This could help provide circuit designers with a better understanding of the design problem. Performance models can also help designers to better formulate the design problems with virtualization, sensitivity analysis, and optimization.

2.2 Performance Model Construction

Technique Approaches Global surrogate modeling [1] is used to create performance models with good accuracy over the complete design space. This is different from building a local surrogate model for the purpose of optimization [2, 3].

Surrogate modeling accuracy and efficiency are determined by several key steps including the sampling plan, model template, and validation.

The first step is determination of a sampling plan. A good sampling plan decides how to efficiently choose samples for fitting good models, considering that the number of samples is limited by the desire to constrain computational expense. Recently, adaptive sampling techniques were developed in order to achieve better efficiency in sampling [4, 5]. These techniques analyze the data from previous iterations and select new samples in the areas that are more difficult to fit.

The model template selection step determines the surrogate model type and model complexity. Multiple popular surrogate model types are available, including rational functions, Kriging models, radial basis function (RBF) models, artificial neural networks (ANNs), and support vector machines (SVMs) [4, 6]. All of these methods can have an embedded analytic coarse model. The model complexity is controlled by a set of hyperparameters which would be optimized during a modeling process.

The model validation step establishes the predictive capabilities of the models and estimates their accuracy. One popular method is fivefold cross-validation [4, 6] in which the training data are divided into five subsets. A surrogate model is then constructed five times: each time four subsets are used for model construction and one subset is used for error measurement. The model error can be measured as an absolute error, e.g., maximum absolute error (MAE), or a relative error, e.g., root mean square error (RMSE), root relative square error (RRSE), and Bayesian estimation error quotient (BEEQ).

Automatic Modeling Flow Figure 2 presents an automatic modeling flow that is able to generate performance models from transistor-level circuit simulations. Before the modeling starts, sets of input and output parameters are defined. The modeling techniques are also configured, including the model template, adaptive sampling strategy, and accuracy measurement. An accuracy target is defined as well. At the beginning of the modeling process a small set of initial samples are generated. Then transistor-level Simulation Program with Integrated Circuit Emphasis (SPICE) simulations are performed using this initial set, and the corresponding responses are collected and used as the modeling data. Surrogate models are then constructed and their parameters optimized. The model accuracy is measured, and the optimization continues until only negligible improvements can be made by changing the model parameters. If the desired accuracy is not reached, the adaptive sampling is evoked to add a new set of samples. The process continues until the fit reaches the targeted accuracy. When the process finishes, the model expressions are exported and used in the follow design steps.

In the examples presented in this section, the modeling techniques are explored using the SUrrogate MOdeling (SUMO) Matlab Toolbox [7]. SUMO is a plug-in based, adaptive platform that can be customized flexibly. The toolbox makes it feasible to test a variety of modeling techniques. Transient circuit simulators, including Cadence Virtuoso Spectre® and Synopsys HSPICE®, are used here for performing transistor-level circuit simulations.

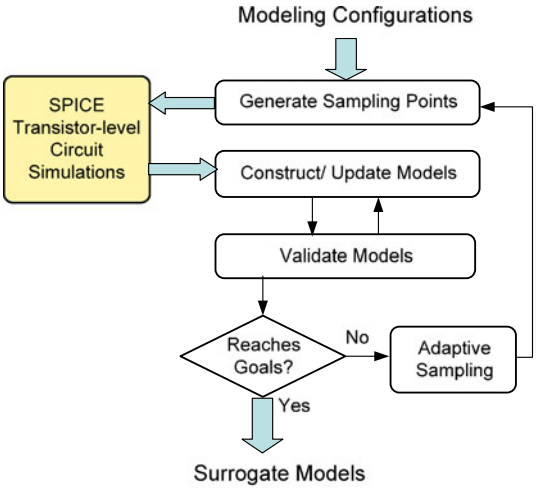


Fig. 2 Automatic adaptive circuit performance surrogate modeling flow

2.3 Surrogate-Based Design and Optimization

Once the surrogate models are constructed, it is possible to explore the multidimensional design space by plotting the model surfaces. The design space set by different design parameters and parametric variation values can be conveniently explored. By using the models it is possible to quickly estimate design feasibility. It is also feasible to compute global sensitivity using the models, and to obtain the relations among factors such as the performance versus the design parameters, the performance versus the variation parameters, and the correlation among the design and variation parameters.

Surrogate models can be integrated to enable global circuit optimization which requires a great number of iterative evaluations of objective functions. In the surrogate-based optimization process there are generally two types of simulation models: a low-fidelity and a high-fidelity model. In circuit design problems the transistor-level circuit simulation is used as a high-fidelity model while the constructed surrogate model is used as the low-fidelity model. Figure 3 shows a general surrogate-based optimization process for circuit design. This method can accommodate additional samples chosen as the optimal design is approached, with the guidance of the existing surrogate model based on past samples [3]. As the surrogate model is used as a guide for adaptive sampling, model accuracy is enhanced with a higher density of samples near the optimum design. Some optimization flows do not involve model updating. Therefore, a global surrogate model is fitted and used as a surrogate for the expensive functions. This method requires high-fidelity surrogate models and uses a relatively large number of samples to build the models. Adaptive sampling can also be used during initial model construction, to obtain better sampling efficiency.

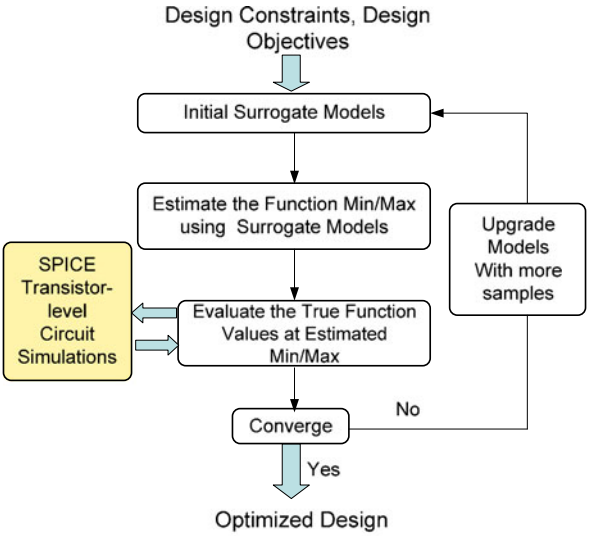


Fig. 3 Surrogate-based optimization flow

2.4 Application Case: Self-Calibrated Low Noise Amplifier Design

This section presents self-calibrated circuit design with application of the surrogate-based performance modeling and model-based optimization.

Recently, post-silicon calibration techniques have been used to compensate for the impact of process-voltage-temperature (PVT) variations on analog and RF circuits [8, 9]. With appropriate design, the circuits are capable of self-calibrating their performance. However, a general design infrastructure and the tools to assist a self-calibrated design are unavailable. Without a general design method, it takes much *ad hoc* work to realize such an adaptive design.

Recently, a new design approach was developed for designing cost-effective self-calibrating analog and RF circuits [10]. The key idea is building multiple operating states to compensate for large PVT variations. In particular, each operating state tolerates a specific range of variations. In circuit implementation a set of design parameters are selected to construct the operating states, and their values for each state are determined at the design phase. A surrogate model-based design flow was developed to select these effective design parameters and to design the optimal operating states.

This section presents a self-calibrated low noise amplifier (LNA) circuit design using this new design approach. The 12 GHz LNA design with the simplified circuit schematic shown in Fig. 4 was implemented in a 65 nm CMOS process. The LNA employs a cascode structure which is composed of a common-source amplifier stage followed by a common-gate amplifier stage. A negative g_m cell is used in the load. The supply voltage is 1.2 V. The specifications of interest are voltage gain, noise factor (NF), $|S_{11}|$, and power. Table 1 lists the nominal performance and

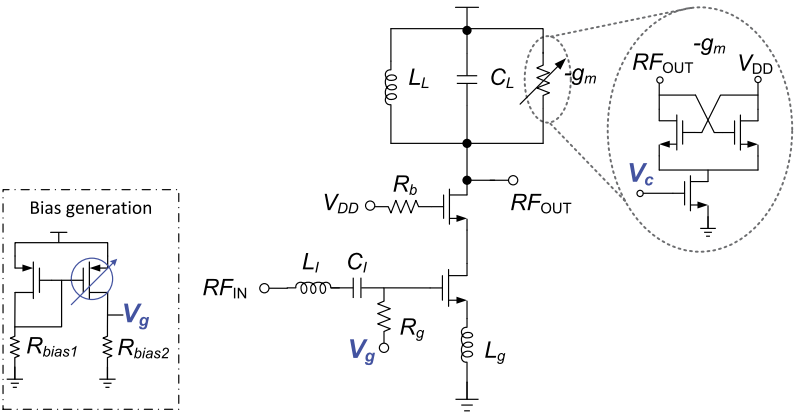


Fig. 4 Simplified schematic of the low noise amplifier (LNA) circuit [10]

Table 1 Performance specifications (center frequency = 12 GHz) [10]

Performance	Specification	Nominal performance
Gain (dB)	≥20	22.46
$\|S_{11}\|$ (dB)	≤−15	−18.10
NF (dB)	≤2.8	2.17
Power (mW)	≤35	25

design specifications. Two key bias voltages, the input bias voltage V_g and tail bias voltage V_c in the negative g_m cell, are selected as the tunable design parameters for self-calibration. These bias voltages are chosen as they provide critical control over the performance of interest and introduce low area overhead. To ensure that the transistors are working in the correct operating regions, the tunable bias V_g has the range 0.58 V to 0.8 V and V_c has the range 0.58 V to 1.0 V. As a design choice, the bias voltages V_c and V_g can be tuned in 10 mV steps, provided by the voltage generator, indicated in Fig. 4.

In this case, we consider transistor threshold voltage variation ΔV_{th} as the main process variation and assume that ΔV_{th} has a normal distribution with $3\sigma = 15$ %. Temperature is considered as an additional environmental variation and is in the range of −10 °C to 80 °C. The variation space is uniformly partitioned into 36 subregions. This requires that each operating state be able to tolerate 5 % threshold voltage variation and 15 °C temperature change.

Model Construction The automatic modeling process discussed in Sect. 2.2 was used to construct the performance models. A Kriging model [4] was selected as the model template. The Gaussian correlation function and a linear regression were used in the kriging model construction. The hyperparameters of the models were optimized using a genetic algorithm [7] suitable for the cases where little problem specific information can be explored.

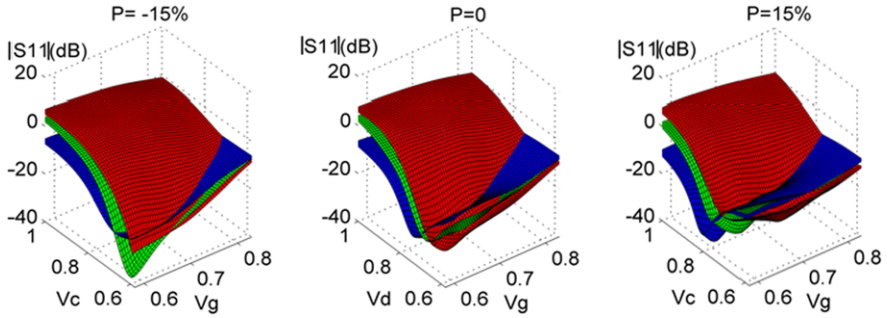


Fig. 5 Slice plots of performance response $|S_{11}|$. The three slices in each plot are for three temperatures: 80 °C, 45 °C, and −10 °C

Table 2 LNA performance modeling samples and accuracy measurement with RRSE and MRE [10]

Performance	Samples	RRSE	MRE
Gain	257	3.50 %	6.7e-2
$\|S_{11}\|$	459	4.39 %	7.1e-2
NF	180	3.01 %	5.7e-2
Power	102	1.54 %	3.33e-3

An initial optimized Latin hypercube design [4] of size 24 was used augmented with the corner points. New samples were selected using the adaptive sampling algorithm with a combination of LOLA-Voronoi [5] and error-based sampler. LOLA-Voronoi is a highly adaptive sampling algorithm which performs a trade-off between exploration (filling up the design space as equally as possible) and exploitation (selecting data points in highly nonlinear regions). Error-based sampling is driven by the evaluation of the model on a dense grid, and this algorithm selects samples in the locations where the model error is estimated to be the largest. Fivefold cross-validation [4, 6] was used to assess the model accuracy. In addition to the training set, a separate data set of 500 samples was available for validation purposes. This data set was sampled by Latin hypercube design and was not used in training the models. The model accuracy was measured using the root relative square error (RRSE) and maximum relative error (MRE).

Figure 5 shows the slice plot of the performance model $|S_{11}|(V_g, V_c, P, T)$. The sample numbers and the achieved accuracy are listed in Table 2. The results show that the modeling cost is dominated by $|S_{11}|$ modeling, as its response is strongly nonlinear.

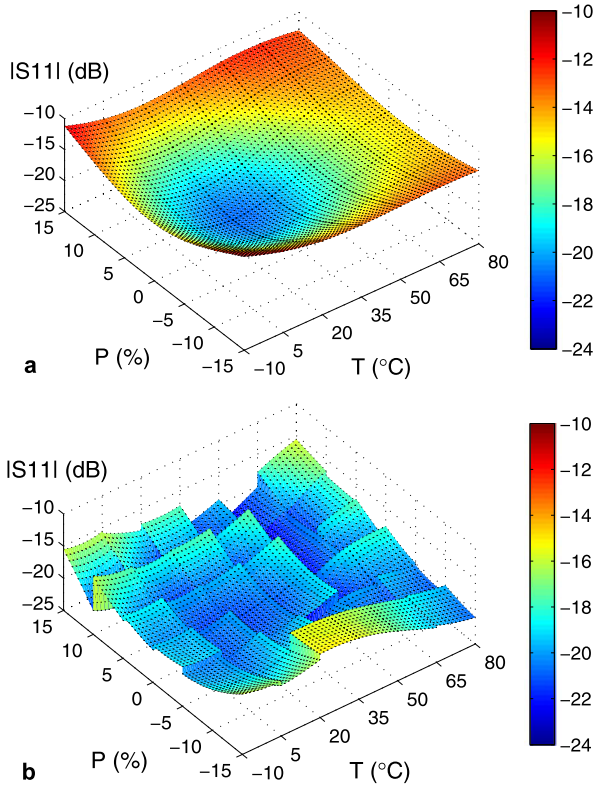


Fig. 6 Plot of performance response $|S_{11}|$: (**a**) the nominal design, (**b**) the self-calibrated design [10]

Model-Based Design The design objective is to minimize the performance deviations from the nominal performance. The design objective function is defined as follows:

$$\begin{aligned} \text{Minimize} \quad & \Phi(\mathbf{X}, \mathbf{V}) = \left\| f_1(X, V), f_2(X, V), f_3(X, V), f_4(X, V) \right\|_2 \\ & f_i(\mathbf{X}, \mathbf{V}) = \frac{P_i(X, V) - P_{i\_\text{norm}}}{P_{i\_\text{norm}}}, \quad i = 1, 2, 3, 4 \end{aligned} \tag{1}$$

where P_i is one of the performance parameters $|S_{11}|$, gain, NF, and power, as selected by the index i. P_{i_norm} is the ith nominal performance. $\mathbf{X}$ is the vector of the tunable design parameters, i.e., bias voltages V_g and V_c. $\mathbf{V}$ is a set of variation conditions used for validation.

Design Results As the tunable bias voltages are discrete variables, a grid search is used with the performance surrogate model and the derived operating states. Figure 6 shows $|S_{11}|$ performance responses of the nominal design and of the self-calibrated design. Figure 6a shows that when the design parameters are set to their

Table 3 LNA performance calculated using transistor-level simulations of the nominal design and the self-calibrated design [10]

Performance specification	Nominal design		Self-calibrated design	
	μ	σ	μ	σ
Gain $\geq$ 20 dB	21.93	1.03	22.62	0.49
$\lvert S_{11}\rvert \leq -15$ dB	-16.89	2.58	-20.03	1.13
NF $\leq$ 2.8 dB	2.32	0.34	2.31	0.27
Power $\leq$ 35 mW	24.23	2.14	21.50	2.8

Table 4 LNA performance yield calculated using transistor-level simulations of the nominal design and the self-calibrated design [10]

Performance yield	Nominal design	Self-calibrated design
Gain	99.8 %	100 %
$\lvert S_{11}\rvert$	66.6 %	100 %
NF	87.5 %	100 %
Power	100 %	100 %
Total	66.6 %	100 %

nominal values, only a limited number of the subregions are able to meet specifications. Figure 6b shows that when the tunable parameters for the optimal design are set in each sub-region, all of the sub-regions are able to meet the performance constraints.

Validation The results are validated using transistor-level Monte Carlo simulations. It is assumed that ΔV_{th} has a normal distribution with a 3σ of 15 % and a uniform temperature distribution from -10 °C to 80 °C. Note that the process variations are bounded by $\pm 3\sigma$. The results after calibration were generated using the precomputed optimal operating states for the subregions. The circuit performance was analyzed by running 1,000 Monte Carlo simulations. Tables 3 and 4 list the performance mean (μ), standard deviation (σ), and yield. As the results show, the performance degradation is fully compensated and thus the final circuit is able to meet specifications.

In summary, the method designs multiple operating states in the LNA circuit in order to compensate for extensive process and temperature variations. By considering both performance constraints and variation tolerance at the design stage, the circuit is able to calibrate performance responding to the variations. A surrogate-based process is constructed to obtain the optimal operating states. Surrogate modeling technologies approximate the circuit performances as a function of design parameters and parametric variations using the results of detailed circuit simulations obtained using a small number of samples. Thus, it efficiently assists design analysis and optimization. The results show that the new design method reduces performance distribution and significantly improves the circuit yield.

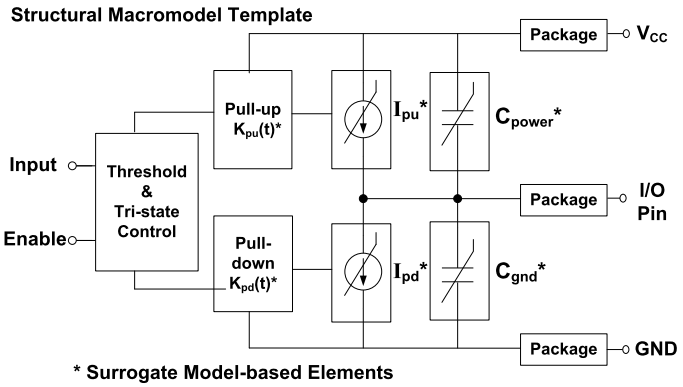


Fig. 7 Structural IO buffer macromodel template with surrogate model elements [12]

3 Accurate and Scalable IO Macromodel

Macromodels of input/output (IO) circuits are essential for fast timing, signal integrity, and power integrity analyses in high-speed digital systems. The most popular approach to IO modeling is to use the traditional table-based input-output buffer information specification (IBIS) [11]. IBIS models are simple, portable, IP-protected, and fast in simulations. However, they are unable to simulate continuous process-voltage-temperature (PVT) variations and are unsuitable for statistical analysis. This section describes a new type of macromodel, called the surrogate IBIS model, to solve this problem [12]. In the new method, an equivalent circuit structure is used to capture the static and dynamic circuit behaviors, while surrogate modeling is used to approximate each element over a range of PVT parameters, so that the macromodel is able to dynamically adapt to the PVT variations in analysis. Figure 7 shows the structure of the surrogate-based IBIS model. I_{pu} and I_{pd} represent the nonlinear output current. Time-variant coefficients K_{pu} and K_{pd} determine the partial turn-on of the pull-up/down networks during switching transitions. C_{power} and C_{gnd} represent the nonlinear parasitic capacitance between the output and the supply rails. Surrogate models of these model elements are constructed to capture the effects of supply voltage, terminal voltages, semiconductor process, and temperature.

3.1 IO Macromodel Construction

The automatic modeling process described in Sect. 2.2 can be used to construct surrogate models for the model elements in Fig. 7. Here modeling data extracted from transistor-level SPICE circuit simulations is used. Figure 8a shows the circuit test-bench used to extract the pull-up output current I_{pu} (V_{S}, V_{pu}, T, ΔV_{th}). The parameter V_{pu} is defined as the voltage difference between the power supply rail and the output, and it ranges from $-V_{\text{CC}}$ to $+2V_{\text{CC}}$, covering the maximum reflection

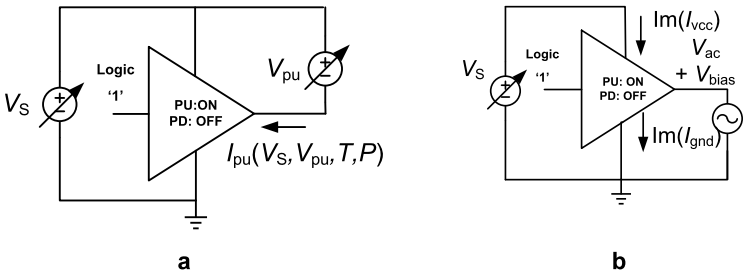


Fig. 8 Test-benches for extracting model elements: (**a**) pull-up current I_{pu}, (**b**) output capacitance C_{gnd} and C_{power} [12]

case [13]. Transient simulations are performed, and the simulation time must be long enough to record a stable output current I_{pu}. Similarly, the pull-down current model I_{pd} was extracted by turning on the pull-down network and turning off the pull-up network. I_{pd} was extracted as a model function of PVT variations and V_{pd}, where V_{pd} is defined as the voltage difference between the output and the ground.

The data was used to fit rational function models in the form

$$f(X) = \frac{P(X)}{Q(X)}, \tag{2}$$

where P and Q are polynomial functions in $X = \{x_1, x_2, \ldots, x_n\}$, and Q is nonzero. P and Q have no common factor of positive degree.

The test setup for extracting the output parasitic capacitance is shown in Fig. 8b. An AC signal is attached to the output ports, and the imaginary currents in the power and the ground ports are measured. The capacitances C_{power} and C_{gnd} were derived using Eq. (3):

$$C_{power} = \frac{\Im(I_{VCC})}{2\pi f V_{AC}}, \quad \text{and} \quad C_{gnd} = \frac{-\Im(I_{gnd})}{2\pi f V_{AC}}, \tag{3}$$

where $\Im(I_{VCC})$ and $\Im(I_{gnd})$ are the imaginary parts of the measured currents, f is the frequency of the AC source, and V_{AC} is the AC voltage amplitude.

The time-variant transition coefficients K_{pu} and K_{pd} were obtained using the 2EQ/2UK algorithm [14]. Figure 9a shows the test to obtain the switching output voltage waveforms. A simplified circuit to illustrate the 2EQ/2UK algorithm is shown in Fig. 9b. The switching output voltage waveforms wfm_1 and wfm_2 were obtained for different terminal voltages V_{term}, and the unknown coefficients K_{pu} and K_{pd} were derived using the equations

$$K_{pu}(t) I_{pu}\left(V_{\mathrm{wfm}_1}(t)\right) - K_{pd}(t) I_{pd}\left(V_{\mathrm{wfm}_1}(t)\right) - I_{out} = 0 \tag{4}$$

and

$$K_{pu}(t) I_{pu}\left(V_{\mathrm{wfm}_2}(t)\right) - K_{pd}(t) I_{pd}\left(V_{\mathrm{wfm}_2}(t)\right) - I_{out} = 0, \tag{5}$$

where $I_{out} = (V_{out} - V_{term})/R_{load}$. I_{pu} and I_{pd} are the output current models.

The new model can be implemented in the Verilog-A behavioral version of the IBIS model [15], in which the surrogate models are implemented in the form of analog functions.

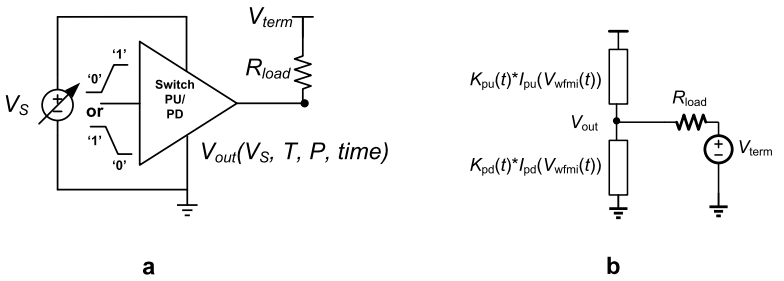


Fig. 9 (**a**) Test-benches for extracting rising/falling transition waveforms for K_{pu} and K_{pd}. (**b**) Illustration of 2EQ/2UK algorithm [12]

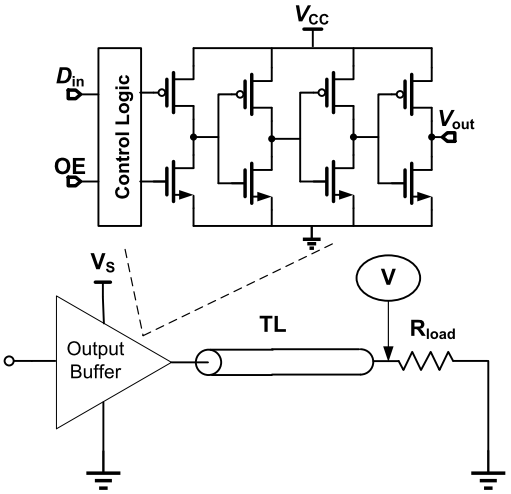


Fig. 10 Simplified buffer circuit and test setup for validation [12]

3.2 Example

This section presents the macromodeling of a single-ended output buffer circuit with the new method. Figure 10 shows the circuit designed in the 180 nm CMOS process with a 3.3 V nominal supply voltage. The threshold voltage variations ΔV_{th} in the MOS transistors are considered as the main process variations, and they are assumed to be within ± 20 % of the nominal value V_{th0}. The parameter $P = \Delta V_{th}/V_{th0}$ is used to describe the threshold voltage variation. The supply voltage V_s is assumed to fluctuate within ± 30 % of the nominal supply (3.3 V), and the temperature (T) is set in the range of 0 to 100 °C. In the modeling process, those PVT-related parameters are sampled adaptively in their ranges.

The generated surrogate IBIS model is compared to the reference provided by the transistor-level simulation, and to the traditional IBIS model extracted from SPICE using the S2IBIS3 v1.0 tool [16].

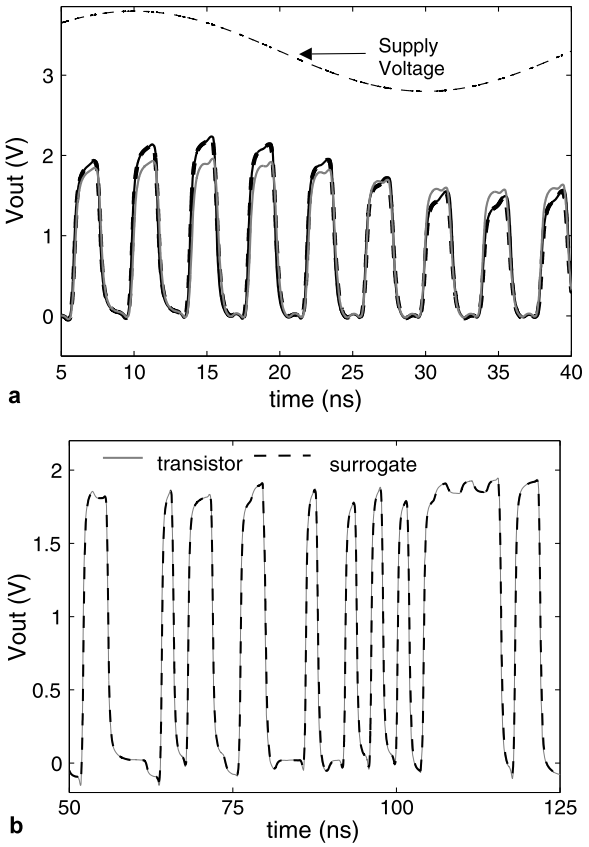


Fig. 11 Output voltage at the far end of the transmission line. (**a**) Case 1, *black solid line*—transistor model, *gray solid line*—traditional IBIS, *black dashed line*—proposed surrogate IBIS. *Black dash-dotted line*—supply voltage. (**b**) Case 2, *gray solid line*—transistor, *black dashed line*—macromodel [12]

The test setup is shown in Fig. 10 where the driver is connected to a 0.75-m long lossy transmission line (RLGC model) with a loading resistor. The characteristic impedance of the transmission line is 50 Ω. The loading resistor is 75 Ω. Two test cases were examined. The results are shown in Fig. 11.

1. Case 1: This case has a 250 MHz square wave as the test input signal. The input data has the pattern "01010" with a 0.1-ns rise/fall time and 2-ns bit period. The supply voltage varies from 2.8 to 3.8 V.
2. Case 2: This case has a data pattern with a 1,024-bit-long pseudorandom bit sequence (PRBS) with 2-ns bit time. The power supply voltage is constant.

The accuracy of the macromodels is quantified by computing the timing error and the maximum relative voltage error. The timing error is defined as the time difference between the reference and the macromodel voltage responses measured for

crossing half of the output voltage swing. The maximum relative voltage error is defined as the maximum error between the reference and macromodel voltage responses divided by the voltage swing.

The results for Case 1 show that when there is a large variation of the supply voltage, the surrogate IBIS model has much better accuracy for both the timing error and the relative voltage error than does the traditional IBIS model. The maximum timing error of the surrogate IBIS model is 79 ps, and the maximum relative voltage error is 6.77 %. The surrogate IBIS model achieves the improved accuracy by capturing the complex output capacitance characteristics and the effects of the supply voltage and gate modulation effects on the output current [17].

In Case 2, the results show that the surrogate IBIS model achieves good accuracy. In this case the maximum timing error is 70 ps (3.5 % of the bit time) and the maximum relative voltage error is 6.45 %. An analysis of the eye diagram of the output in Case 2 is a further test of model fidelity. The eye width (W) was measured when the eye height (H) was equal to 1 V. The results under different PVT conditions show that the eye-width differences are within 0.04 ns (2 % of the bit time).

In summary, the proposed surrogate IBIS macromodel achieves good accuracy in analysis. The macromodels obtained show good accuracy in capturing the effects of reflections and variations, and their scalability enables flexible design analysis.

4 Surrogate-Based Device Modeling

Scaling of device sizes induces high variability of transistor parameters. There are two major reasons for this. First, quantum mechanics-based phenomena such as drain-induced barrier lowering (DIBL) or gate tunneling, which are negligible in long-channel devices, become more significant. Additional physics-based effects increase the dependence of many circuit design quantities (including the drain current, I_{ds}, and device transconductance, g_m) on transistor process parameters such as the oxide thickness, t_{ox}. Furthermore, the tolerances of semiconductor manufacturing components do not scale down as the transistor sizes shrink [18]. As a consequence, the amount of uncertainty of the design quantities remains constant as device sizes become smaller, leading to higher percentage variability with respect to the nominal values of the transistor process parameters. The experimental data revealed that a traditional process corner analysis might not reflect the real distribution of the critical transistor parameters such as the threshold voltage V_{th} [19], while the Monte Carlo analysis becomes more computationally intensive with the increasing number of variability factors.

The response surfaces of design quantities, which become more complex with the process variations, can be accurately captured by surrogate modeling. Surrogate modeling aims to express the output quantity in terms of a few input parameters by evaluating a limited number of samples. These samples are used by the basis functions which establish the response surface of the desired output. The coefficients of the basis functions for the response surfaces should be optimized to minimize

modeling error. This approach has been applied to the problem of I_{ds} modeling in order to assess the effects of variability in analog circuit building blocks, in particular, for differential amplifiers [20]. In this section, the modeling of g_m of n-channel transistors will be discussed.

The transconductance g_m is an important quantity for analog circuits, particularly in determining the AC performance of amplifiers, mixers, and voltage-controlled oscillators. The modeling here is based on 65 nm device technology (IBM 10SF design kit) and uses six process parameters (t_{ox}, intrinsic threshold voltage $V_{th,0}$, intrinsic drain-source resistance $R_{ds,0}$, intrinsic mobility μ_0, channel length variation ΔL_{eff}, and channel doping N_{ch}) as input to the model in addition to the terminal voltages of the transistor (gate-source voltage V_{gs}, drain-source voltage V_{ds}, and bulk-source voltage V_{bs}) and the temperature T. The choice of these process parameters is based on their physical origin, which ensures a weak correlation between each parameter. The I_{ds} equations of the BSIM model are analytically differentiated to yield g_m [21]:

$$g_m = \partial I_{ds} / \partial V_{gs}. \tag{6}$$

The g_m expression is validated by extensive SPICE circuit simulations over the process corners and at temperature extremes so that it can be used to evaluate the samples, each a function of the ten parameters described above. Although an analytic equation for g_m is used in this work, the modeling methodology is general and can employ simulations or measurement results if they have the same input and output parameters.

Kriging basis functions are used to construct the surrogate model, and the necessary coefficients are optimized using the MATLAB toolbox Design and Analysis of Computer Experiments (DACE) [22]. The device width is assumed to be 10 μm. The final model is tested for accuracy using the root relative square error (RRSE) metric [4, 6].

The g_m model was constructed using a total number of 2,560 input samples and tested with 6,400 samples not part of the set of input samples used in developing the model. The resulting model yields an RRSE of 3.96 %, indicating a high level of accuracy.

The model can be used to observe the changes in g_m with respect to its input parameters. Examples of this are provided in Fig. 12. The graphs provide critical insight to the designer about the fundamental relations and trade-offs between the chosen process parameters, terminal voltages, and temperature. Higher g_m values are obtained with smaller V_{th0}, L_{eff}, and t_{ox}, as well as larger μ_0. This information becomes especially vital when variability of the circuit performance that depends on g_m must be considered. In the example of an RF cascode LNA, the voltage gain A_v, input and output scattering parameters S_{11} and S_{22}, as well as the optimum noise impedance Z_{opt} are complex functions of the g_m value of the common source transistor [23]. Any variability of the process parameters of this transistor may push the design outside of the specification range. In this case, the information presented in Fig. 12a–c can be used to change the matching network of the amplifier such that it can yield the desired design metrics in all cases of process variability.

Finally, note that surrogate model-based device modeling is not limited to a single design quantity. Response surface models of other important design metrics can

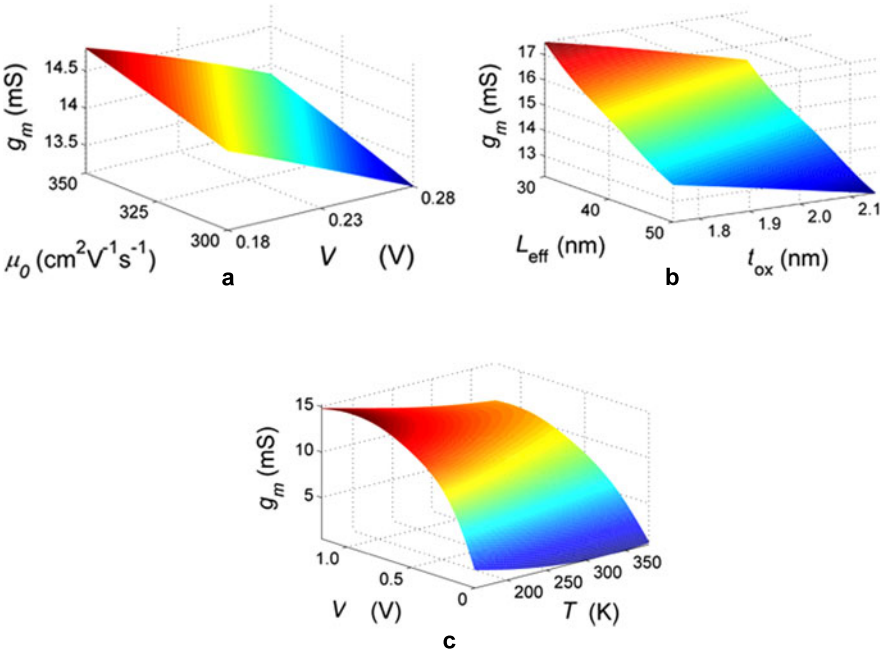


Fig. 12 3D graphs showing the trade-offs between the different inputs on the modeled g_m. (**a**) inputs are μ_0 and V, (**b**) inputs are Leff and t_{ox}, (**c**) inputs are V and T

also be developed by using the methodology described here. As an example, consider the bandwidth of a single-stage amplifier. The bandwidth is a function of both the process parameters used in g_m modeling and of the parasitic capacitances of the transistors. However, these capacitances also depend on some process parameters. The exact relationships can be quantified by analytical expressions as given in the device model equations [21]. Once the additionally required parameters are determined, then the surrogate modeling process can be applied as in g_m modeling.

It should be kept in mind that the main premise of using surrogate models in device modeling is both to enhance the computational efficiency and to provide the designer with insight about the relation of the process parameters and circuit outputs. With respect to the first goal, Monte Carlo methods have been traditionally applied to describe the output variation of circuits. However, Monte Carlo methods require a very large number of samples to be simulated to yield meaningful relations between the process parameters and circuit parameters. Surrogate models can be used to replace Monte Carlo methods by building a response surface model representing the circuit outputs in terms of the input parameters and using it instead of the actual simulations to determine the effects of process variations. Thus, the computational cost will be reduced considerably.

The second goal, which tries to identify the impact of die-to-die and within-die variations on circuit outputs, has been achieved using application-specific process cards that are generated to satisfy different levels of process variations leveraging

from device-level measurements [24]. This method accurately indicates the level of variations in the circuit outputs, but it does not really show how within-die variations can alter the circuit output. Additionally, designers must use multiple different process cards to check the performance variations, which can result in conflicting recommendations to be implemented in design. Surrogate models aim to unify the information regarding the process variations in a single model. The designer can then use this model to visualize both the die-to-die and within-die variations by evaluating different devices with distinct process parameters that are not necessarily specified by a process card.

5 Conclusions

This chapter presented applications of surrogate modeling to variation-aware circuit macromodeling and design analysis. Surrogate modeling facilitates efficient design exploration and optimization with variation-aware performance models. An example was provided that showed that surrogate modeling can be used to enhance the accuracy and scalability of IO macromodels. Moreover, the surrogate model-based method is able to generate device models with critical variability parameters. The surrogate-based method greatly reduces the complexities and costs of variation-aware macromodeling and circuit design.

Acknowledgements The chapter has the distribution statement "A" (Approved for Public Release, Distribution Unlimited). The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. This work was supported by the Self-HEALing mixed-signal Integrated Circuits (HEALICs) program of the Defense Advanced Research Projects Agency (DARPA) and the prime contractor Raytheon Company (Contract number: FA8650-09-C-7925).

References

1. Gorissen, D., Turck, F.D., Dhaene, T.: Evolutionary model type selection for global surrogate modeling. J. Mach. Learn. Res. **10**(1), 2039–2078 (2009)
2. Eldred, M.S., Dunlavy, D.M.: Formulations for surrogate-based optimization with data fit, multifidelity, and reduced-order models. In: 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Portsmouth, Virginia (2006). AIAA-2006-7117
3. Koziel, S., Cheng, Q.S., Bandler, J.W.: Space mapping. IEEE Microw. Mag. **9**(6), 105–122 (2008)
4. Forrester, A., Sobester, A., Keane, A.: Engineering Design via Surrogate Modeling: A Practical Guide. Wiley, Chichester (2008)
5. Crombecq, K., Couckuyt, I., Gorissen, D., Dhaene, T.: Space-filling sequential design strategies for adaptive surrogate modelling. In: First International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering. Civil-Comp Press, Stirlingshire (2009)
6. Yelten, M.B., Zhu, T., Koziel, S., Franzon, P.D., Steer, M.B.: Demystifying surrogate modeling for circuits and systems. IEEE Circuits Syst. Mag. **12**(1), 45–63 (2012)

7. Gorissen, D., Crombecq, K., Couckuyt, I., Dhaene, T., Demeester, P.: A surrogate modeling and adaptive sampling toolbox for computer based design. J. Mach. Learn. Res. **11**, 2051–2055 (2010)
8. Han, D., Kim, B.S., Chatterjee, A.: DSP-driven self-tuning of RF circuits for process induced performance variability. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **18**(2), 305–314 (2010)
9. Natarajan, V., Hunag, H., Makris, Y., Drineas, P.: BIST driven power conscious post-manufacture tuning of wireless transceiver systems using hardware-iterated gradient search. In: Asian Test Symposium (ATS), November 2009, pp. 243–248 (2009)
10. Zhu, T., Steer, M.B., Franzon, P.D.: Surrogate model-based self-calibrated design for process and temperature compensation in analog/RF circuits. IEEE Des. Test Comput. (2012). doi:10.1109/MDT.2012.2220332
11. IO Buffer Information Specification [Online]. http://www.vhdl.org/ibis/. Accessed 2 Feb 2009
12. Zhu, T., Steer, M.B., Franzon, P.D.: Accurate and scalable IO buffer macromodel based on surrogate modeling. IEEE Trans. Compon. Packag. Manuf. Technol. **1**(8), 1240–1249 (2011)
13. IBIS Modeling Cookbook. http://www.vhdl.org/pub/ibis/cookbook/cookbook-v4.pdf. Accessed 1 May 2009
14. Muranyi, A.: Accuracy of IBIS models with reactive loads. http://www.eda.org/pub/ibis/summits/feb06/muranyi2.pdf. Accessed 7 July 2009
15. LaBonte, M., Muranyi, A.: IBIS Advanced Technology Modeling Task Group Work-achievement: Verilog-A element library HSPICE test. http://www.eda.org/ibis/macromodel_wip/archive/20070323/mikelabontecisco/. Accessed 7 July 2009
16. Varma, A., Glaser, A., Lipa, S., Steer, M.B., Franzon, P.D.: The development of a macro-modeling tool to develop IBIS models. In: 12th IEEE Topical Meeting on Electrical Performance Electronic Packaging, Princeton, New Jersey, pp. 277–280 (2003)
17. Varma, A.K., Steer, M.B., Franzon, P.D.: Improving behavioral IO buffer modeling based on IBIS. IEEE Trans. Adv. Packaging **31**(4), 711–721 (2008)
18. Orshansky, M., Nassif, S.R., Boning, D.: Design for Manufacturability and Statistical Design: A Constructive Approach. Springer, New York (2008)
19. Saha, S.K.: Modeling process variability in scaled CMOS technology. IEEE Des. Test Comput. **27**(2), 8–16 (2010)
20. Yelten, M.B., Franzon, P.D., Steer, M.B.: Surrogate model-based analysis of analog circuits—Part I: variability analysis. IEEE Trans. Device Mater. Reliab. **11**(3), 458–465 (2011)
21. Morshed, T.H., Yang, W., Dunga, M.V., et al.: BSIM4.6.4 MOSFET model user's manual. http://www-device.eecs.berkeley.edu/bsim/Files/BSIM4/BSIM470/BSIM470_Manual.pdf (2009). Accessed 1 October 2012
22. Lophaven, S.N., Nielsen, H.B., Sondergaard, J.: A MATLAB Kriging toolbox 2.0. http://www2.compute.dtu.dk/~hbni/dace/dace.pdf (2002). Accessed 1 June 2009
23. Yelten, M.B., Gard, K.G.: A novel design methodology for tunable low noise amplifiers. In: Wireless and Microwave Conference (WAMICON'09), Florida, pp. 1–5 (2009)
24. Lin, C., Dunga, M.V., Lu, D.D., Niknejad, A.M., Hu, C.: Performance-aware corner model for design for manufacturing. IEEE Trans. Electron Devices **56**(4), 595–600 (2009)

Computing Surrogates for Gas Network Simulation Using Model Order Reduction

Sara Grundel, Nils Hornung, Bernhard Klaassen, Peter Benner, and Tanja Clees

Abstract CPU-intensive engineering problems such as networks of gas pipelines can be modelled as dynamical or quasi-static systems. These dynamical systems represent a map, depending on a set of control parameters, from an input signal to an output signal. In order to reduce the computational cost, surrogates based on linear combinations of translates of radial functions are a popular choice for a wide range of applications. Model order reduction, on the other hand, is an approach that takes the principal structure of the equations into account to construct low-dimensional approximations to the problem. We give an introductory survey of both methods, discuss their application to gas transport problems and compare both methods by means of a simple test case from industrial practice.

Keywords Reproducing kernels · Radial basis functions · Model order reduction · Proper orthogonal decomposition · Gas transport · Networks · Differential algebraic equations

1 Introduction

Gas network simulation is a challenging problem that can become large scale, if one considers a realistic network. However, the ability to simulate these large models accurately and efficiently is becoming increasingly more important as gas is used more widely as an energy source. In order to optimize processes related to gas pipeline operation or other gas network problems, it is important to be able to simulate the flow of gas within complex pipe networks in real time. This is a challenging problem, since the flow within a pipe is a complex physical phenomenon and the network itself can be very large and complex, spanning entire continents

S. Grundel (✉) · P. Benner
Max Planck Institute, Magdeburg, Germany
e-mail: sara.grundel@mpi-magdeburg.mpg.de

N. Hornung · B. Klaassen · T. Clees
Fraunhofer SCAI, Sankt Augustin, Germany

N. Hornung
e-mail: nils.hornung@scai.fraunhofer.de

S. Koziel, L. Leifsson (eds.), *Surrogate-Based Modeling and Optimization*,
DOI 10.1007/978-1-4614-7551-4_9,

with pipelines joined by numerous junctions. When we write the system as a set of equations for discrete values of the pressure and flows or possibly other important states, the system size can increase to an order of tens of thousands. Since the equations are highly nonlinear differential algebraic equations, this is already a challenging problem. Creating surrogate models in general can often solve the problem. Since we are interested in solutions of equations, the functions to be modelled are only given implicitly. Model order reduction is however a technique that is used to reduce the complexity of dynamical systems or sometimes parameter-dependent equations. The reduction of the dimension will typically reduce the computational time. It possibly may allow us to compute a solution, where standard solvers fail due to the complexity of the system. One of the problems for gas networks is for example to compute a stationary solution for a given fixed configuration of supplies and demands within a network. Reduced-order modelling is particularly powerful when used during an optimization algorithm where one needs to solve the equations several times. If a trust region method is used, one is even able to give at least heuristic error estimators, or bounds. In certain cases it can be interesting to create a surrogate of the solution vector by interpolation methods. One of these methods is radial basis metamodelling. Radial basis function (RBF) metamodelling is a very powerful technique, since it can be used to interpolate any given smooth data set.

In Sect. 2 we give a detailed explanation of the modelling of gas within a pipe. We explain the benchmark equations often used as well as the approximations we will apply for modelling the whole network. We show the effects of this approximation in Sect. 5.1. We summarize in Sect. 3 the radial basis function interpolation methods. Section 4 on model order reduction gives a very brief general introduction into model order reduction and a summary of the model order reduction technique we used, which is called proper orthogonal decomposition. We will show first results in Sect. 5 on reducing two different ways of modelling the gas network. One is a dynamic model and the other a stationary one. In the stationary case we also compare it to a radial basis surrogate in which we can see the advantage and limitations of this model.

2 Gas Network Simulation

Modelling and simulation of a gas network is a challenging problem for many reasons. One is that gas pipe network simulation connects combinatorial aspects, through the network topology, with continuous behaviour within individual components. The physics within these components is often complex, and one needs to find suitable approximations. The gas flow within one individual pipe is described by nonlinear partial differential equations (PDEs). When we add valves, for example, the solution of the system is no longer differentiable. Because the problem can be arbitrarily complex, we will have to limit ourselves within this chapter. We cannot address all the challenges of gas networks, but rather mean to describe and discuss certain aspects of a simplified, though still practically relevant, version of the problem.

We will mainly follow [8, 13, 23] for the mathematical modelling of the gas network. We are going to discuss a dynamical model and a quasi-stationary model to describe the gas flow within a pipe. The topology of a pipe network, after [23], is modelled as a directed graph $\mathscr{G} = (\mathscr{A}, \mathscr{N})$, where the vertices are supply nodes $\mathscr{N}_+$, demand nodes $\mathscr{N}_-$ and interior nodes $\mathscr{N}_0$ (junctions),

$$\mathscr{N} = \mathscr{N}_+ + \mathscr{N}_- + \mathscr{N}_0.$$

The edges of the graph can constitute pipes, connections, compressors, valves, regulators or other components of a realistic gas network. We will, however, restrict ourselves to a network that consists of pipes only. This is certainly a strong simplification, but the main focus of this chapter is to develop a fast surrogate for the problem. One can still explain the techniques of surrogate modelling we use as well as address issues within the simulation of gas networks, namely the problem of finding a stationary solution for a given configuration.

An edge is denoted by $ij \in \mathscr{A}$, where the flow is directed from i to j. Within each pipe we want to model the physics of the gas by thermodynamic conservation laws. This procedure is based on [8].

We now assume that $x \in [0, L]$ is the independent variable along the length of a pipe segment, where L is the length of the segment (D is its diameter). The other independent variable, if one considers time-varying problems, is the time t.

Conservation of mass yields what is called the continuity equation (1) and conservation of momentum the pressure loss equation (2). These equations form a coupled PDE together with the equations of state of a real gas (3). Altogether they are called the isothermal Euler equations:

$$\partial_t \rho + \partial_x q = 0, \tag{1}$$

$$\partial_t q + \partial_x p + \partial_x \left(\rho v^2\right) + g\rho\partial_x h = -\frac{\lambda(q)}{2D}\rho v|v|, \tag{2}$$

$$p = \gamma(T) z(p, T)\rho. \tag{3}$$

The dependent state variables are gas density $\rho(x,t)$, gas velocity $v(x,t)$, gas pressure $p(x,t)$ and gas temperature $T(x,t)$. From the density and velocity of the gas we can calculate the gas flow $q(x,t) = \rho(x,t)\, v(x,t)$. The network of pipes can have a given geodesic height $h(x)$. The friction coefficient $\lambda(q)$ can be determined by a variety of formulæ. The simplest one is

$$\lambda = 0.067 \left(\frac{158}{\mathrm{Re}} + \frac{2k}{D}\right)^{0.2},$$

where Re is the Reynolds number and $k = 0.03$ mm the equivalent roughness of the pipe [16]. If we even consider an approximation of Re, λ will be a constant number independent of q. The term $z(p,T)$ is a compressibility factor that can also be computed by several formulæ [16]. It denotes the deviation from the behaviour of an ideal gas ($z = 1$) and may even be approximated by this ideal behaviour without

the loss of practical utility of the equations. $\gamma = RT$ is a field determined by temperature and a gas constant R [16]. The full term γz is often approximated by the square of the sound velocity $a \approx 300$ m/s in practice. Making use of the described simplifications yields

$$\begin{aligned} \partial_t \rho + \partial_x q &= 0, \\ \partial_t q + \partial_x p + \partial_x \left(\rho v^2\right) + g\rho\, \partial_x h &= -\frac{\lambda}{2D} \rho v |v|, \\ p &= a^2 \rho. \end{aligned}$$

We now consider a small pipe network and allow for an approximation of $h(x) \equiv h_0$ and $T(x) \equiv T_0$. We will also drop the kinetic energy term, replace $v = \frac{q}{\rho}$, plug the third equation into the second and obtain:

$$\begin{aligned} \partial_t \rho + \partial_x q &= 0, \\ \partial_t q + a^2 \partial_x \rho &= -\frac{\lambda}{2D} \frac{q|q|}{\rho}. \end{aligned} \tag{4}$$

We will consider two models for our pipe network, one where the physics within the pipe is modelled by (4) and the other where we assume the dependent variables ρ and q are time-invariant as in [13, Sect. 2.3], leading to the quasi-static equations

$$\partial_x \rho = -\frac{\lambda}{2Da^2} \frac{q|q|}{\rho}, \quad q = \text{const.} \tag{5}$$

Equations (4) or (5) must be satisfied for every pipe $ij \in \mathscr{A}$. The pipes are connected at the nodes and for each interior node we have to satisfy Kirchhoff's current law; i. e. the sum of all incoming gas flows equals the sum of all outgoing flows per node. In the following, we will describe how to discretize these two sets of equations inside a given network and see the full set of equations to be solved. We start with (4) which is the dynamic case and use these equations to then derive the quasi-static discretized equation in Sect. 2.2. This quasi-static case uses the equations for the stationary solution. If one considers time-dependent inputs, however, one can use these simplified equations to construct time-dependent functions for q and ρ. This is a relevant approach in practice and is referred to as the quasi-static solution.

2.1 Dynamic Equations

Given a network with pipes as edges, experience shows that in practice any rough discretization of $x_i - x_{i-1} \in [1{,}000\text{ m}, 5{,}000\text{ m}]$ is sufficient. We therefore take discrete points for the pressure only at the nodes and flows q at the beginning and at the end of each pipe. If the pipe is longer than 5 km we add artificial interior nodes

into the network. We then have a density ρ_i at every node $i \in \mathscr{N}$, and for every pipe segment $ij \in \mathscr{A}$ we have a flow q_{ij}^{out} and a flow q_{ij}^{in}. Here (out) stands for the flow at the end of the pipe, out of the pipe, and (in) for the flow at the beginning of the pipe. The dynamic pipe equations are then discretized for a given pipe ij by

$$\partial_t \frac{\rho_i + \rho_j}{2} + \frac{q_{ij}^{\text{out}} - q_{ij}^{\text{in}}}{L_{ij}} = 0,$$
$$\partial_t \frac{q_{ij}^{\text{out}} + q_{ij}^{\text{in}}}{2} + a^2 \frac{\rho_j - \rho_i}{L_{ij}} = -\frac{\lambda}{4D_{ij}} \frac{(q_{ij}^{\text{out}} + q_{ij}^{\text{in}})|q_{ij}^{\text{out}} + q_{ij}^{\text{in}}|}{\rho_i + \rho_j}.$$

For every node we have to add an algebraic constraint. Introducing the new variable $y_{ij} = \frac{q_{ij}^{\text{out}} + q_{ij}^{\text{in}}}{\rho_i + \rho_j}$, one can write the full system as:

$$\partial_t \frac{\rho_i + \rho_j}{2} + \frac{q_{ij}^{\text{out}} - q_{ij}^{\text{in}}}{L_{ij}} = 0 \quad \forall e = ij \in \mathscr{A}_{\text{pipe}}, \tag{6}$$
$$\partial_t \frac{q_{ij}^{\text{out}} + q_{ij}^{\text{in}}}{2} + a^2 \frac{\rho_j - \rho_i}{L_{ij}} + \frac{\lambda}{4D_{ij}} \left| q_{ij}^{\text{out}} + q_{ij}^{\text{in}} \right| y_{ij} = 0 \quad \forall e = ij \in \mathscr{A}_{\text{pipe}}, \tag{7}$$
$$\sum_{ji \in \mathscr{A}} q_{ji}^{\text{out}} - \sum_{ik \in \mathscr{A}} q_{ik}^{\text{in}} = 0 \quad \forall i \in \mathscr{N}_0, \tag{8}$$
$$\sum_{ji \in \mathscr{A}} q_{ji}^{\text{out}} - \sum_{ik \in \mathscr{A}} q_{ik}^{\text{in}} - D_i(t) = 0 \quad \forall i \in \mathscr{N}_-, \tag{9}$$
$$a^2 \rho_i(t) - \hat{p}_i(t) = 0 \quad \forall i \in \mathscr{N}_+, \tag{10}$$
$$(\rho_i + \rho_j) y_{ij} - \left(q_{ij}^{\text{out}} + q_{ij}^{\text{in}} \right) = 0 \quad \forall e = ij \in \mathscr{A}_{\text{pipe}}. \tag{11}$$

Equations (6) and (7) describe the dynamics of the system. Equation (8) is Kirchhoff's current law for interior nodes. In (9) the sum of flows has to be equal to the demand at the demand node. For the supply node, we have a given pressure $\hat{p}_i(t)$ (10). Equation (11) is necessary since we introduce the extra variables y_{ij}.

We will consider $\lambda, 1/D_{ij}, 1/L_{ij}, a^2$ as parameters and collect them in a large vector p. The demand flows $D_i(t)$ and the supply pressures $\hat{p}_i(t)$ are considered inputs to the system and are collected in a vector $u(t)$. The states of the system are $\rho_i, q_{ij}^{\text{out}}, q_{ij}^{\text{in}}, y_{ij}$. Collecting all states in a vector x, this system can be written as

$$E\dot{x} = A(p)\,x + H(p)\left(x \otimes g(x)\right) + Bu, \tag{12}$$

where $(g(x))_i = x_i$ or $(g(x))_i = |x_i|$. In a variety of applications one is interested in optimizing a quantity that depends on the states. If one calls that quantity $y = h(x)$ we are in the framework of a dynamical system of the form

$$E\dot{x} = A(p)\,x + H(p)\left(x \otimes g(x)\right) + Bu,$$
$$y = h(x).$$

Due to the fact that E is singular, this system is called a differential-algebraic equation (DAE). For general network structures it is an index 1 DAE. Furthermore, it is reasonable to assume $g(x) = x$ for certain networks. Then the dynamical system is purely quadratic. The matrices in (12) are sparse and the parameter dependence is linear.

2.2 Quasi-static Equation

Starting from Eq. (5) we follow [13, Eq. (6)] for an explicit solution, which is a discretization of the pipe with nodes at the beginning and end of the pipe only:

$$\begin{aligned}
\partial_x \rho &= -\frac{\lambda}{2Da^2}\frac{q|q|}{\rho}, \quad q = \text{const}, \\
\Rightarrow \rho\partial_x \rho &= -\frac{\lambda}{2Da^2} q|q|, \quad q = \text{const}, \\
\Rightarrow \frac{1}{2}\partial_x\left(\rho^2\right) &= -\frac{\lambda}{2Da^2} q|q|, \quad q = \text{const}, \\
\Rightarrow \partial_x\left(p^2\right) &= -\frac{a^2\lambda}{D} q|q|, \quad q = \text{const}, \\
p_j^2 - p_i^2 &= -\frac{a^2\lambda}{D_{ij}} q_{ij}|q_{ij}|L_{ij}, \quad q_{ij} = \text{const}.
\end{aligned}$$

Together with Kirchhoff's laws at the nodes the full algebraic system is given by

$$p_j^2 - p_i^2 + \frac{a^2\lambda}{D_{ij}} q_{ij}|q_{ij}|L_{ij} = 0 \quad \forall e = ij \in \mathscr{A}, \tag{13}$$

$$\sum_{ji\in\mathscr{A}} q_{ji} - \sum_{ik\in\mathscr{A}} q_{ik} = 0 \quad \forall i \in \mathscr{N}_0, \tag{14}$$

$$\sum_{ji\in\mathscr{A}} q_{ji} - \sum_{ik\in\mathscr{A}} q_{ik} - D_i = 0 \quad \forall i \in \mathscr{N}_-, \tag{15}$$

$$p - \hat{p}_i = 0 \quad \forall i \in \mathscr{N}_+. \tag{16}$$

With the parameters $a^2, \lambda, L_{ij}/D_{ij}$ collected in p, the pressures and flows (p_i, q_{ij}) collected in x, and $\hat{p}_i$ as well as D_i collected in u we get

$$A(p)\,x + H(p)\left(x \otimes g(x)\right) + Bu = 0, \tag{17}$$

where $(g(x))_i = |x_i|$ or x_i again. In some cases one can assume the relevant x to be positive as in the dynamic case. Then $g(x) = x$ and the system is quadratic.

2.3 Discussion of Nonlinearity

The gas flow within the pipe network leads to a moderate nonlinearity of the quadratic term and possibly to a discontinuity due to the absolute value. Nevertheless, further elements like connections, valves and regulators may introduce a stronger nonlinearity or can even cause a loss of smoothness within the solution. Connections do not introduce nonlinearity, since they can be thought of as simple, very short pipes with a constant relative pressure loss. If valves are seen as either open or closed, they will also not introduce any further nonlinearity. An opening or closing procedure of a valve would cause a discontinuity within the state and should therefore be modelled differently, i.e. by a smooth transition. The remaining elements, the regulators or control valves, do introduce strong nonlinearity, since they will usually reduce gas pressure, but they do not interfere if the gas pressure is too low to be reduced. One must ensure that the solution algorithm of the DAE can deal with such non-smoothness, either by appropriate adaptation of the algorithm itself, or by a smoothing procedure. The character of these kinds of nonlinearities or discontinuities should be studied further. Within the scope of this chapter and with model order reduction in mind, we will consider the moderately nonlinear part only, since regulators will not essentially contribute to the size of the system.

3 Radial Kernel Interpolation

Before we give a detailed overview of model order reduction of gas transport problems, we introduce a more general interpolation approach for comparison. Radial basis function (RBF) interpolation is a very intuitive black-box interpolation approach that is easy to implement and makes very general assumptions about the function that is to be approximated. Recall, for example, the quasi-static equations in matrix form (17). If we use an RBF approach, we do not consider the state x, but only the implicitly defined function of parameters p onto one (or more) outputs y,

$$f : p \in \Omega \subset \mathbb{R}^d \to y \in \mathbb{R}.$$

However, if no output is defined, one can consider the whole state vector as output. The assumptions of RBF interpolation about the approximand f are, simply put, continuity and smoothness. Since we are dealing explicitly with an implicitly defined function, we cannot make these assumptions in general. On the other hand, we are dealing with a mildly nonlinear setting, where in practice we can expect continuity and smoothness most of the time. The RBF interpolant is then given by the linear combination of M translates of a radial function $\phi(r)$, $r \in \mathbb{R}^+$,

$$s : x \in \Omega \subset \mathbb{R}^d \to \sum_{i=1}^{M} \lambda_i \phi\big(\|x - x_i\|\big), \tag{18}$$

for given interpolation points $x_1, \ldots, x_M$. Here, we adopt the convention from the RBF literature, where x denotes a parameter from Ω, while in the model order reduction literature x denotes a state. $\|\cdot\|$ stands for the 2-norm or Euclidean norm of a d-dimensional vector. Although other norms have been used, many results in the literature are obtained for this specific choice; refer to [6], [9, Chap. 10] for a brief discussion of alternatives. The coefficients λ_i are determined by M interpolation conditions

$$s(x_j) = f(x_j) \quad \forall j \in [1, \ldots, M] \tag{19}$$

of a sampling $f(x_j), x_j \in \mathscr{X} \subset \Omega$ of the parameter space Ω. The choice of this sampling strongly determines the quality of the approximation. Since the interpolation is only dependent on Euclidean distances, scattered samplings that do not lie on a regular grid can be considered without any difficulty.

Classical choices of RBFs are infinitely differentiable functions such as (inverse) multiquadrics $\phi(r) = (\varepsilon^2 r^2 + 1)^\beta$, $\beta \in \mathbb{R}$ or Gaussians $\phi(r) = e^{-\varepsilon^2 r^2/2}$. The shape parameter $\varepsilon > 0$ scales the width of the functions and thus influences the quality of the approximation critically. Other important choices of RBFs are finitely differentiable, such as the Matérn class of radial functions and the polyharmonics [5] as well as radial functions with compact support. As a prominent example among compactly supported radial functions is that of the Wendland functions stand out [9, 24]. Wendland functions speed up the setup and evaluation of the interpolant, thereby trading computational expense for accuracy. Since RBF interpolants can always be set up and evaluated comparably fast within our framework of the gas transport problem, we do not consider radial functions with compact support in the following, nor will we report on any other accelerating approaches.

3.1 Convergence Rates

In order to be able to derive general properties of RBF interpolation, RBFs are seen as symmetric positive definite reproducing kernels $K(x, y)$ that induce certain Hilbert spaces [19]. The main idea is to choose the kernel K as the translate of a radial function

$$K(x, y) = \phi\big(\|x - y\|\big).$$

Then the theory of reproducing kernels can be applied and error estimates as well as convergence rates of RBF interpolation can be concluded. The Hilbert space induced by a kernel is called its 'native' Hilbert space, and properties of the kernel such as the radial symmetry and its translation invariance are responsible for the innate structure of this induced space. Not strictly positive definite functions such as the popular multiquadrics can be incorporated into the framework as conditionally positive definite kernels. Polynomial detrending, which allows us to exactly

reproduce polynomial functions of a low degree, plays an important role in industrial practice as well as in the theory of conditionally positive definite reproducing kernels. We will use polynomial detrending and conditionally positive definite kernel interpolation within this chapter, but without a thorough theoretical treatment. The reason is that we are mainly concerned with the basic practical questions of its application to gas network problems. The interested reader is referred to the introduction of the topic in [19] as well as to the very comprehensible and practical treatment in [9] and to the comprehensive account of all the theory involved that is given in [24].

Our main assumptions of continuity and smoothness, as we mentioned earlier, can be seen in terms of properties of the native space. If the approximand f lives in the native space of an RBF, the literature provides upper bounds for error estimates depending on the fill distance $h = \sup_{x\in\Omega} \min_{x_j\in\mathscr{X}} \|x - x_j\|$ of a sampling $\mathscr{X}$. Here h is a quantity that measures how well a sampling $\mathscr{X}$ covers the parameter space Ω. These error bounds typically are given such that they have a factor which depends on h and a second factor which depends on $|f|_{\mathscr{N}_K(\Omega)}$, where $|\cdot|_{\mathscr{N}_K(\Omega)}$ is the norm of the native space of the kernel K. In other words, the given error bounds try to separate the influence of the sampling from that of the approximand f. Both factors do depend on the kernel K, though—a fact which assigns an important role to our choice of the radial function ϕ. A general rule of thumb is that we can obtain better convergence rates with respect to h if the approximand is infinitely smooth and thus can be approximated by an infinitely smooth RBF interpolation. Gaussians and multiquadrics allow for exponential convergence orders in the best case if we keep ε constant [9, Chap. 15], while the known best bounds for finitely smooth radial functions such as the Matérn class of functions and thin plate splines are polynomial. The Matérn functions

$$\phi(r) = \frac{K_{\beta-d/2}(r)\, r^{\beta-d/2}}{2^{\beta-1}\Gamma(\beta)}, \quad \beta > d/2$$

have Sobolev spaces $W_2^\beta(\mathbb{R}^d)$ as their native spaces $\mathscr{N}_\phi(\mathbb{R}^d)$ [9, Chap. 13] with

$$W_2^\beta\left(\mathbb{R}^d\right) = \left\{g \in L_2(\Omega) \cap C(\Omega) : D^\alpha g \in L_2(\Omega)\ \forall |\alpha| \le \beta,\ \alpha \in \mathbb{N}^d\right\},$$

where Ω is bounded and D^α denotes the distributional derivative with multi-index α. The native spaces of Gaussians and multiquadrics are smaller. If we try to approximate a function f which lives in $W_2^\beta(\mathbb{R}^d)$ but outside the native space of a multiquadric or Gaussian, the expected convergence rate becomes polynomial again, even if we apply a multiquadric or Gaussian kernel, and is therefore comparable to the convergence of finitely smooth RBF interpolation [18]. Although their native space is rather small, we will therefore mainly use multiquadrics in our experiments.

3.2 Shape Parameter Selection

Apart from the important choice of the type of radial basis, there are several other parameters to consider that influence the approximation quality. The most obvious is the scaling parameter ε, which is also called the shape parameter or 'width' in the literature and which determines if the translate of a basis function has a more local or global influence. The discussion about the choice of ε is very similar to that of the basis function itself. Some researchers therefore argue that very flat and thus smooth and global functions yield the most accurate approximations [10, 11].

Unfortunately, flat basis functions cause numerical ill-conditioning; thus high-accuracy results cannot be obtained with the described standard method. The solution of the system of interpolation conditions (19) leads to large coefficients λ_i introducing numerical cancellation errors within the interpolant; recall (18). If this ill-conditioning grows too strong, the theoretically favourable convergence rates are lost [9, Chap. 17].

Several remedies have been suggested in the literature, such as a computationally more expensive expansion of the interpolant s by a rational polynomial of finite order [10, 12]. Other authors use numerical preconditioning of the system or certain basis transformations [9, Chap. 34], [17]. In practice, the problem of ill-conditioning becomes important if the approximand f exhibits strong local nonlinearity, while most of the function is rather smooth. If this is the case, the simplest solution is to locally adapt the shape parameter so that it takes a different value for each basis function. The goal is to use smoother basis functions in smoother areas of the approximand and less smooth basis functions in more nonlinear regions. Such an adaption of ε, which was originally proposed in [15], is easy to implement and potentially even enlarges the native space of infinitely smooth RBFs (it does not for the finitely smooth Matérn class functions and thin plate splines [5]). Unfortunately, one cannot ensure that the system is still non-degenerate when varying ε, although this method has often been used in practice with good results [11]. If degeneration of the system is an issue, we like to point out some very similar approaches like non-stationary multilevel iteration [9, Corr. 32.1] and adaptive univariate interpolation by scaled multiquadrics [4]. The details of the selection of ε in these methods are beyond the scope of this chapter. The simplest rule of thumb is that the variable shape parameter should decrease in a 'stationary way', i.e. according to the decrease in the fill distance h. The simple gas transport problems considered in this chapter do not make it necessary to apply any of these measures at all. Also, the example on p. 289 of [9, Table 32.6] shows that a simple uniform choice of ε can still give good approximation results compared to a multilevel approach. However, if a problem exhibits strong local nonlinearities, we recommend trying one of the adaptive or multilevel approaches.

3.3 Sampling and Refinement

A different, though related, question is how to select the samples in $\mathscr{X}$. Since the error bounds are given in terms of the fill distance h, it is intuitive to select $\mathscr{X}$ such that h is minimized. Random or quasi-random samplings (Halton points) are most common. Other empirical choices of $\mathscr{X}$ consider numerical stability or local nonlinearity of the approximand f. Since all these topics have already been discussed in the context of the choice of ε, we omit most of the discussion here for brevity and refer to [9] for more details. The most important result is that non-uniform samplings may be necessary to approximate locally nonlinear functions f, but may cause ill-conditioning. The remedies are essentially again preconditioning, basis transformations, iterative refinement of the sampling and local adaption of the shape parameter such as in multilevel iteration.

Among the black-box surrogate methods, RBF interpolation can be compared to kriging. Reference [20] makes a comparison between intrinsic kriging and general conditionally positive definite kernel interpolation. As stated in [20], radial kernels correspond to kriging with covariance functions of a weakly stationary and isotropic random field.

4 Model Order Reduction (A Special Form of Surrogate)

Model order reduction (MOR) can be seen as a form of surrogate modelling which was originally developed in systems and control theory for dynamical systems. It is neither a black-box approach nor a physical surrogate. MOR can be understood as an interpolatory method that takes the general form of the function that should be approximated into account. Today it is a wide area of research with many applications. For a first introduction into the topic of MOR consult [1, 3, 21].

The main idea behind MOR is that there is a redundancy within the system. The dynamical system, for example, has a large state space, but most of the solution trajectories lie on a lower dimensional subspace. The idea is to solve a system only on that lower dimensional subspace, therefore reducing the size of the state space.

We will introduce the concept of projection-based MOR as a surrogate for dynamical systems of the following problem type:

$$\begin{aligned} E\dot{x} &= Ax + f(x) + Bu, \\ y &= Cx + h(x), \end{aligned} \tag{20}$$

where $u(t)$ is an input function, $x(t)$ is the state vector and $y(t)$ the output function. The idea of projection-based MOR is to find projection matrices $V, W \in \mathbb{R}^{n\times r}$ with $W^T V = I$ and $r < n$ such that $x \approx V\hat{x}$ where $\hat{x} \in \mathbb{R}^r$ solves the reduced equations

$$\begin{aligned} \hat{E}\dot{\hat{x}} &= \hat{A}\hat{x} + \hat{f}(\hat{x}) + \hat{B}u, \\ \hat{y} &= \hat{C}\hat{x} + \hat{h}(\hat{x}), \end{aligned} \tag{21}$$

where $\hat{E} = W^T E V$, $\hat{A} = W^T A V$, $\hat{B} = W^T B$, $\hat{C} = CV$, $\hat{f}(\hat{x}) = W^T f(V\hat{x})$, $\hat{h}(\hat{x}) = h(V\hat{x})$. With this step we have reduced the number of states in the system (the unknowns) as well as the number of equations. The projection we described above is called a Petrov–Galerkin projection if $V \neq W$ and a Galerkin projection if $V = W$.

In order to measure the quality of the reduced system we need a measure for the error. One can measure the error by $\|x - V\hat{x}\|$ for an appropriate norm. This is the focus of proper orthogonal decomposition (POD), which we will explain in more detail in Sect. 4.1. One can also measure the error in the output. When measuring the error in the states we are focusing on the input-to-state map, whereas in the second case in which we measure the error in the output we are interested in the input-to-output map. Other widely used MOR methods are Krylov subspace methods, moment matching, balanced truncation and the reduced basis method.

A problem in projection methods for general nonlinear systems is that, even if one can reduce the number of states and equations considerably, it could still be very expensive to evaluate $\hat{f}(\hat{x}) = W^T f(V\hat{x})$ since in the evaluation of this process one has to create the full large state vector $V\hat{x}$ and evaluate the large vector-valued function f to then project it back down. For linear, quadratic or other more simple systems this additional evaluation of the full system can be avoided as certain matrices can be precomputed. For general nonlinear functions, on the other hand, one has to use the empirical interpolation method (EIM) or the discrete empirical interpolation method (DEIM) [2, 7] which we will not explain here, since the considered gas network problems show only mild nonlinearity.

Different MOR methods have different guarantees on the error created by replacing the original model with the reduced model. Some methods have strict error bounds and others only heuristic error estimators. However, not so much is known for nonlinear systems and especially for nonlinear differential algebraic systems. Another big challenge of nonlinear DAEs is that reducing the system can lead to a system that has no solution. Therefore, there are still many issues to be solved that need further theoretical investigations.

4.1 Proper Orthogonal Decomposition (POD)

We are interested in reducing the system (20). Since POD tries to find a good approximation in the state, we are not interested in the output behaviour. POD finds a projection matrix W to minimize $\|x - W\hat{x}\|$ for a given norm $\|\cdot\|$ at least approximately. POD is a Galerkin projection method, which means we only have to find one matrix W. The method we describe is sometimes referred to as the method of snapshots [22]. Within some communities the general underlying concept of POD is also called principal component analysis (PCA) or Karhuenen–Loeve decomposition. Given samples $x_1, \ldots, x_N$ of any kind (they could be solutions or outcomes of a process in general), the method extracts a basis $u_1, \ldots, u_\ell$ that solves the following

minimization problem:

$$\min_{u_1,\dots,u_\ell} \quad \sum_{k=1}^{N} \left\| x_k - \sum_{i=1}^{\ell} \langle x_k, u_i \rangle u_i \right\| \quad \text{s.t.} \quad \langle u_i, u_j \rangle = \delta_{ij}. \tag{22}$$

The solution to this problem is directly connected to the singular value decomposition (SVD) of the matrix $Y = [x_1, \dots, x_N]$. Given the SVD of $Y = U \Sigma V^T$ the solution to (22) for the standard Euclidean inner product and Euclidean norm is obtained by the first ℓ left singular vectors which are the first ℓ columns of U. Here we assume that the singular values in Σ are ordered. Per definition of the SVD, U and V are orthogonal matrices and Σ is a diagonal matrix.

For a dynamical system as given above, what we would really like to do is solve the following optimization problem:

$$\min \quad \int_0^T \left\| x(t) - \sum \langle x(t), u_i \rangle u_i \right\|^2 \mathrm{d}t \quad \text{s.t.} \quad \langle u_i, u_j \rangle = \delta_{ij}. \tag{23}$$

Since this is difficult and we do not always know the solution $x(t)$ everywhere in time, we solve the approximated version of this. Here we pick times $t_1, \dots, t_N$ and look at solutions at these discrete time steps $x(t_1), \dots, x(t_N)$, the so-called snapshots. We typically consider a fixed input function $u(t)$. The optimization problem (23) reduces to (22).

Given the singular vectors u_i of the snapshot matrix $Y = [x(t_1), \dots, x(t_N)]$ we have that

$$x(t_k) \approx \sum_{i=1}^{\ell} \langle x(t_k), u_i \rangle u_i = \sum_{i=1}^{\ell} \hat{x}_i(t_k) u_i = W \hat{x}(t_k),$$

where W is given by the matrix $W = [u_1, \dots, u_N]$. Since this is true for all discrete time steps t_k, it can be considered as evidence that it is true at arbitrary times, which gives us

$$x(t) \approx W \hat{x}(t).$$

We are not able to compute $\|x(t) - W\hat{x}(t)\|$ at every t, but we minimize

$$\sum_k \left\| x(t_k) - W \hat{x}(t_k) \right\|.$$

If the system is furthermore dependent on a parameter, this is typically extended by creating snapshots for several parameter values and times. All the snapshots will be put into one large matrix Y and the rest can be done as described above.

5 Surrogates for Gas Network Modelling

In this section we will discuss approaches to verify the discussed simplifications for the pipe equations, the different surrogate models constructed by MOR methods and also RBF surrogates.

We will start with a gas network consisting of one pipe only in which we compare the quasi-static approximation with the dynamic approximation and the simulation of the full isothermal Euler equations. We will see the advantages and limitations of our two approaches. Furthermore, we discuss in detail the solution of a larger example network using the quasi-static equations in the pipe. We construct a surrogate with POD MOR as well as with an RBF metamodel. The third test is a simulation of the dynamic model on that given network and its surrogate, computed by POD.

5.1 Pipe Equation Comparison

As a first verification of the pipe equations we are considering one long pipe ($L =$ 100 km) with given pressures at the inlet and outlet. This example was studied in [13]. For comparability we use the same constants $a = 377.9683$ m/s, $\lambda = 0.011$, $D = 0.5$ m. The pressure at the inlet will always be given by constant 45 bar and the pressure at the outlet varies smoothly from $p_O(0) = 40$ bar to $p_O(T) = 45$ bar. We have different test cases with fast and slowly changing inlet pressure profiles as well as one oscillating profile.

1. $p_O(t) = p_O(0) + \frac{p_O(T)-p_O(0)}{2}(1 - \cos(\pi \frac{t}{T}))$ for $T = 1, 3$ and 6 h
2. $p_O(t) = p_O(0) + \frac{p_O(T)-p_O(0)}{2}(1 - \cos(4\pi \frac{t}{T}))$ for $T = 12$ h

We compute the flow at the inlet and outlet by using three different sets of equations. First we solve the benchmark model, the isothermal Euler equations given by (1)–(3). These equations are discretized using a second-order relaxed method with adaptive characteristics speeds; see [13] and the references therein. In [13] the underlying grid has 2,001 spatial points and uses a Courant–Friedrichs–Lewy (CFL) condition of 0.49. We do not adhere to this choice of discretization and construct a graph out of the pipe by adding 19 nodes such that the pipe segments all have a length of 5 km. This, as discussed above, is the maximal length we allow for our network. Apart from the full isothermal equations we also solve the DAEs given by (6)–(11). Then the third solution is the quasi-static solution given by solving Eqs. (13)–(16). One can solve them on the network as above with the 19 added nodes or directly on the whole pipe. The solution will be the same since the flow in the beginning and end of the pipe (or pipe segment) is always the same.

Figures 1 and 2 show a comparison of the isothermal equations, our slightly simplified dynamic discretized equation solved in time by an implicit Euler as well as the quasi-static solution for the pressure profiles (1). One can see of course that the quasi-static solution is the same at the inlet and the outlet and is non-smooth at the final time. The approximated dynamic solution shown in red is closer to the

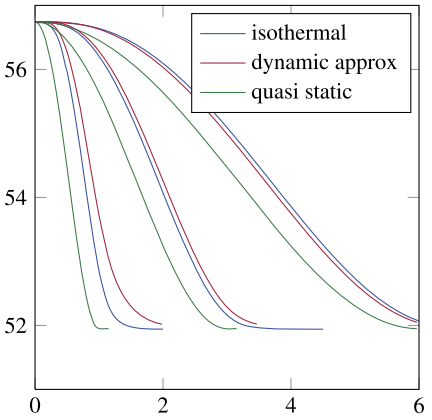


Fig. 1 Case (1). Flux at the inlet

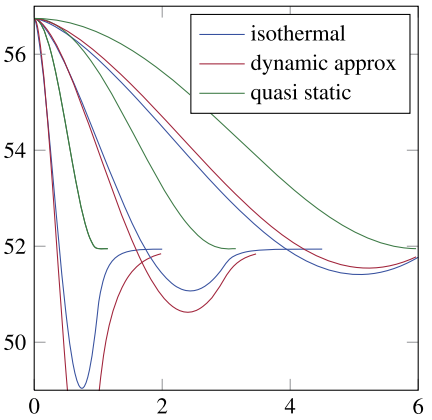


Fig. 2 Case (1). Flux at the outlet

isothermal benchmark solution the more slowly we change the pressure at the inlet. However, it is significantly better than the quasi-static solution which is used often in practical applications. Figures 3 and 4 compare the two different models for oscillating boundary pressure.

In general, one can see that at the inlet the quasi-static solution is a decent approximation, where at the outlet it is not. For slowly varying pressure profiles the dynamic approximation is a very good approximation. This simplification produces a system that is quadratic (if set up accordingly to ensure $g(x) = x$) and can therefore be approximated more efficiently by MOR methods than the full isothermal equations, which justifies its use.

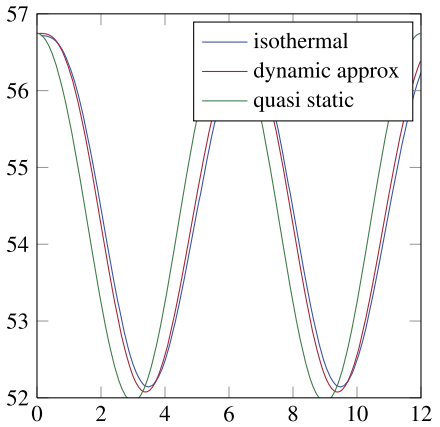


Fig. 3 Case (2). Flux at the inlet

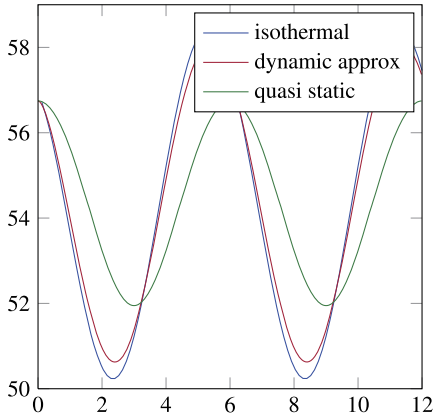


Fig. 4 Case (2). Flux at the outlet

5.2 Test Example

We will compute the surrogate in the two following sections based on a model that is a modified version of a part of a true gas network. It consists of 17 nodes, 16 pipes, 1 supply node and 8 demand nodes. Its graph structure can be seen in Fig. 5. The lengths of the pipes are given by

$$L_{1,2} = 46 \text{ m}, \quad L_{2,3} = 7 \text{ m}, \quad L_{3,4} = 3{,}080 \text{ m}, \quad L_{4,5} = 4{,}318 \text{ m},$$
$$L_{5,6} = 323 \text{ m}, \quad L_{5,7} = 790 \text{ m}, \quad L_{7,8} = 1{,}820 \text{ m}, \quad L_{7,9} = 1{,}460 \text{ m},$$
$$L_{9,10} = 2{,}368 \text{ m}, \quad L_{10,11} = 1{,}410 \text{ m}, \quad L_{11,12} = 296 \text{ m},$$
$$L_{11,13} = 3{,}979 \text{ m}, \quad L_{13,14} = 119 \text{ m}, \quad L_{13,15} = 3{,}881 \text{ m},$$

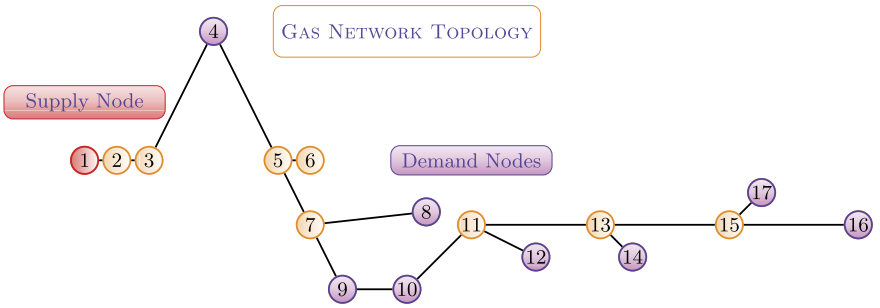


Fig. 5 Gas network graph

$$L_{15,17} = 687 \text{ m}, \qquad L_{15,16} = 6{,}114 \text{ m}$$

and the diameters of the pipes are a constant 0.206 m. We will also assume $\lambda = 0.0003328$ and $a = 430.5\,\text{m/s}$ in the following if not specified otherwise. Furthermore, the supply pressure is 44.5 bar and the demands are given by:

$$\begin{aligned} D_4 &= 0.21 \text{ kg/s}, & D_8 &= 34.86 \text{ kg/s}, & D_9 &= 0.22 \text{ kg/s}, \\ D_{10} &= 2.83 \text{ kg/s}, & D_{12} &= 1.81 \text{ kg/s}, & D_{14} &= 1.04 \text{ kg/s}, \\ D_{16} &= 2.85 \text{ kg/s}, & D_{17} &= 1.45 \text{ kg/s}. \end{aligned}$$

However, we will also consider a trivial input vector where all demands are zero and the supply (or generally all supplies) are a constant p_0. Then setting all pressures to p_0 and all flows to 0 is always a trivial solution of the equations. Since even computing stationary solutions for arbitrary demand and supply distributions can be difficult, one sometimes has to use this kind of trivial solution as a starting configuration.

5.3 *Quasi-static Case with POD*

The equation is given by (17), which we recall here for convenience:

$$A(p)\,x + H(p)\left(x \otimes g(x)\right) + Bu = 0. \tag{24}$$

The parameter p consists of $\lambda, a^2, D_{ij}, L_{ij}$ and the input u of the pressure at the supply node and the outflows at the demand nodes. Since we are mainly interested in computing stationary solutions with this equation the question is: Can one create a good surrogate to find the solution x for a given u? In general applications one typically knows the parameters to be in some range or with some uncertainty. Given the example network described above, we compute solutions $x_0, \ldots, x_N$ for

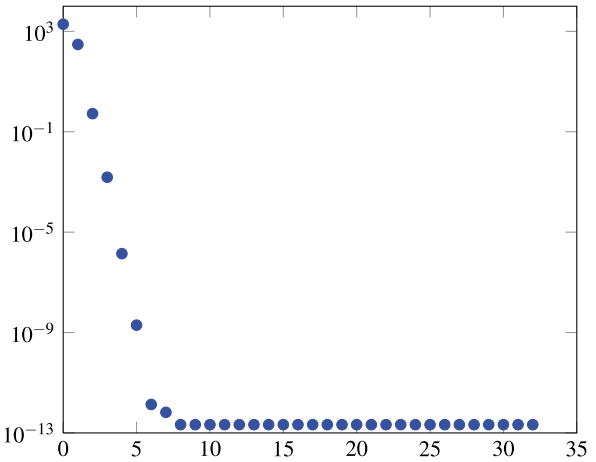


Fig. 6 Singular values

different inputs $u_0, \ldots, u_N$ with

$$u_i = u_0 + \frac{i}{N}(u_N - u_0).$$

Here u_0 for the trivial solution is a starting input as described above with $p_0 \equiv 45$ bar. This means we are solving the equation for a path of input distributions that move from the trivial solution u_0 to the one for u_N, which is the desired input as described in Sect. 5.2. We are using the equation solver `fsolve` from [14] to solve the system above. The singular values of the snapshots matrix $Y = [x_0, \ldots, x_N]$ are given by Fig. 6.

This shows that there is a low-dimensional linear space in which the solutions lie approximately. We create the reduced model by the projection described in Sect. 4.1. This is a special case, since we only have to consider algebraic equations, but no differential equations. However, we consider a time-varying input system and compute the stationary solution of the system for discrete time steps. Interpreting this sequence of stationary solutions as a time-dependent state, we talk about the quasi-static solution, which can be a reasonable solution as we have seen above if the inputs vary slowly enough. Concretely, given the projection matrix W from the snapshot matrix Y by POD, the reduced equation looks like

$$W^T A W\hat{x} + W^T H(p)\,(W \otimes I)\big(\hat{x} \otimes g(W\hat{x})\big) + W^T Bu = 0.$$

Given a tolerance of 10^{-8} we create a reduced model of order 5. Having obtained that reduced model, we compute solutions for u_{TEST} which has the same demands as u_N, but the supply is given by 50 bar. Furthermore, we vary the parameter λ slightly around its given value. The error $\|x - W\hat{x}\|$ is given in Fig. 7 for the 1,000 different test values of $\lambda \in [0.0002, 0.0006]$.

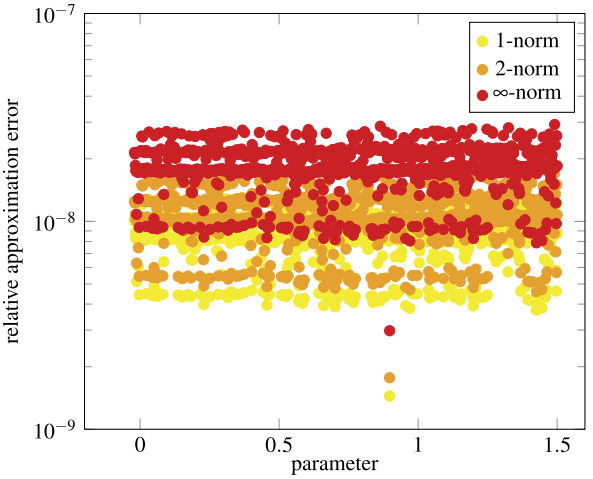


Fig. 7 Different error norms

This shows that, for a given path of input vectors, starting from a trivial configuration, the system is reducible without producing a significant error. Given a configuration u for which the original system is hard to solve, one can create a path $u_0, \ldots, u_N = u$ and solve the system starting with u_0 for each u_i, where `fsolve` is started with the solution of the previous problem. It is known that one can still run into a point where the Newton method no longer converges. One can then use the information from before to reduce the system, solve the reduced system and get a more accurate starting value for the quasi-Newton solver. If we are interested in parameter studies for a given input, the above example already shows that the reduced model is a good model for many parameters. However, we could also reduce the model based on snapshot solutions for different parameters. We computed a snapshot matrix for values of λ varying from 0.0002 to 0.0006, choosing 40 equally spaced points in between. The reduced model is then of order 4, using the same tolerance of 10^{-8} in the singular values. Given 1,000 values of λ between 0 and 1.5, the maximum error of the solution in the reduced system with the solution of the true system is then found to be 10^{-6}.

We also created several RBF metamodels for the solution to this algebraic equation. In order to do that we understand the solution x as an implicitly given function of the parameters and/or the inputs. Multiquadrics, Gaussians and Matérn-class functions have been implemented within a simple interpolation setting. We have not made use of polynomial detrending. Polynomial interpolation is expected to be very efficient and accurate in our example problems, since these problems show a certain polynomial behaviour. On the other hand, such an assumption can probably not be generalized, which is why we refrain from this choice. The shape parameter ε has been selected manually as $\varepsilon = 0.1$. In order to allow for a comparison with POD, the same uniform sampling was used for both methods.

Table 1 Comparing RBF surrogates and POD reduced models

$\lambda \times a^2$	# of samples	Order reduced model	Error RBF	Error POD
$[1.5 \times 10^{-4}, 4.5 \times 10^{-4}] \times [150, 450]$	33	4	10^{-7}	10^{-6}
$[0, 6 \times 10^{-4}] \times [0, 600]$	33	4	2×10^{-6}	10^{-6}
$[0, 6 \times 10^{-4}] \times [0, 600]$	100	5	3.5×10^{-7}	1.6×10^{-7}

In Table 1 we compare creating a surrogate model for the function from the parameters $\lambda \times a^2$ to the state vector x containing the pressures and the flows. In each test we pick a certain parameter range and compute the solution for a number of grid-like distributed points. With that we create an RBF metamodel as well as a reduced-order model via POD as above, where the order is again picked by the tolerance of 10^{-8}. We show the maximum error between the approximated and the true solution sampled on 10,000 grid points.

One can see that if enough samples are chosen within a small enough region, one can get almost arbitrarily small errors in the RBF model. However, this model is only accurate for the exact region in which we have interpolation points. We again stress that we have assumed that the function x of p specified by an implicit function is smooth, which cannot be guaranteed in general. The advantage of the reduced model is that it takes this implicit definition into account and gives good solutions even outside the range which was used to create the model. We can furthermore change other parameters and still get reasonable approximations. On the other hand, it can still be considered a disadvantage that in order to obtain x there is still a system to solve, although of reduced order.

5.4 Dynamic Case

We now consider the set of equations given by (6)–(11). Since the flow is no longer constant within the pipe and we add another set of variables, we have for the example problem a state space dimension of $65 = 16 \times 3 + 17$. Since this is a DAE, we use POD exactly as described above, where we consider the solution $x(t)$ given by solving the system for the input $u(t)$ which has constant demands and a varying supply pressure $\hat{p}_1(t)$ given by

$$\hat{p}_1(t) = 44.5\,\text{bar} + 2.5\,\text{bar} \times \left(1 - \cos\left(\frac{\pi t}{1\,\text{h}}\right)\right), \quad t \in (0, 1.5\,\text{h}).$$

We take 100 snapshots of the solution trajectory $x(t)$ at equidistant time steps for $t \in [0, 1.5\,\text{h}]$. The singular values of the snapshot matrix are given by Fig. 8. Cutting off at 10^{-8}, we create a reduced-order model of size 39. To compare we compute the solution at 3,000 time steps between 0 and 1.5 h. The maximal relative error between the reduced solution and the true solution is then found to be 10^{-8}. This

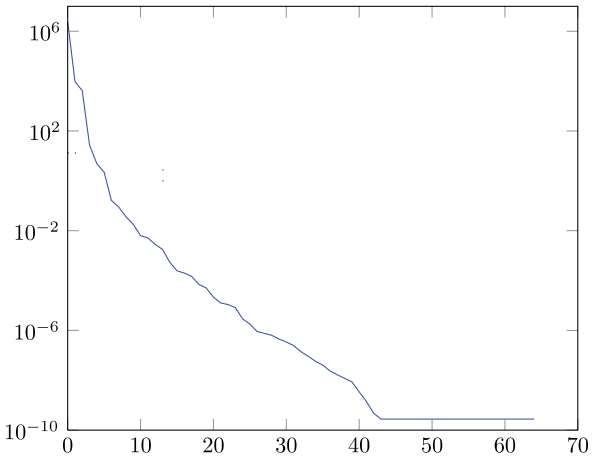


Fig. 8 Singular values of snapshot matrix

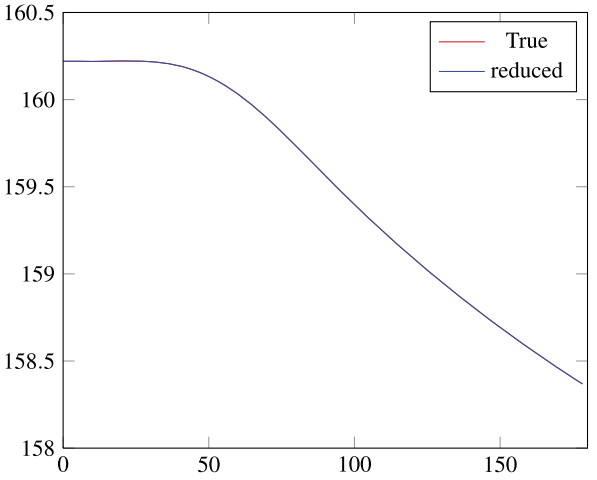


Fig. 9 Flux at node 5

small relative error is not surprising, since cutting off the singular values at 10^{-8} is cutting them off at a relative error of $10^{-8}/10^{6} = 10^{-14}$, where 10^{6} is the value of the largest singular value. And since we reduce the model only for a given input function and fixed parameters, the solution trajectory lies in that subspace and we can reduce the system to that subspace without creating an error. We can see the flux of the original model and the flux of the reduced model at node number 5 in Fig. 9. It shows that there is no difference.

6 Discussion and Conclusions

We have compared a dynamic approximation to the isothermal Euler equations for a network consisting of one long pipe. We use different outlet pressures and can see that the dynamic approximation is very close to the true isothermal Euler equations, especially if the pressure at the outlet is varying slowly enough. The often-used quasi-static approximation is shown to not exhibit the right behaviour, but for very slowly varying inputs it could still be a valuable option. We also investigated a part of a realistic gas network with 17 nodes and 16 pipes. We created a reduced model for the quasi-static set of equations by POD. We were able to reduce the model for varying inputs to just four unknowns with a relative error of only 10^{-7}. This shows clearly that gas networks are reducible and that model reduction is a valued surrogate. We also created RBF surrogates for this problem. Given good sampling points in the whole range, one can create very accurate and efficient RBF interpolants. However, the reduced model gives good results even outside the sampled parameter values. As discussed in Sect. 4, there are many methods for reduced-order modelling; however, very little is known for nonlinear differential algebraic systems. We tested the POD approach, since it is easy to use and it works for arbitrary systems. The selected RBF approach works well due to the simple relation between x and p in our example, but it might not be applicable to more involved settings.

With respect to the dynamic problem, proper orthogonal decomposition is probably not the most efficient model order reduction scheme. $\mathscr{H}_2$-optimal methods for linear systems that allow for a best approximation in the $\mathscr{H}_2$-norm of the system can be computed efficiently by an iterative Krylov subspace algorithm without solving the full system of equations. They have recently been extended to differential algebraic equations and to quadratic problems. However, there are still some issues. It is not yet fully clarified how optimality can be achieved with these methods. Neither is it understood in detail how and when the iterative procedure converges and how to implement the approach efficiently. Network systems have already been studied as nonlinear DAEs and there exist some tractability and stability results. However, we believe that there are still a lot of possible research topics, in order to arrive at tractable and stable reduced-order systems of nonlinear gas transport network problems. One of the main questions is: Does the reduced-order system still adhere to the algebraic equations? Another one is: How can quadratic terms and switched state systems be incorporated into some of the DAE-specific linear model order reduction approaches?

As to RBF interpolation, again, the dynamic case is more involved. There are very general RBF interpolation methods for PDEs that may have the potential to be adapted for gas networks. One might consider time as a parameter or a set of solutions for discrete time steps as a vector-valued output. In the computer graphics literature there are also RBF methods for the interpolation of implicit functions, which might have the potential to be extended to a multidimensional setting and thus to gas network simulation. Of course, the techniques for local strong nonlinearities mentioned in Sect. 3 may be of use if the problem setting becomes more demanding.

As has been mentioned earlier, the gas transport network problem can become much more difficult to solve if nonlinear compressors, regulators or other elements

are considered. Therefore, the model order reduction problem can also become very ambitious. RBF interpolation might deal well with smooth nonlinearities but is expected to have difficulties with piecewise continuous functions or switched state systems. Therefore, one of the challenges in this application area is to combine the different approaches in such a way as to benefit from the advantages of each without suffering their disadvantages.

References

1. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. SIAM, Philadelphia (2005)
2. Barrault, M., Maday, Y., Nguyen, N.C., Patera, A.T.: An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. C. R. Math. Acad. Sci. **339**(9), 667–672 (2004). doi:10.1016/j.crma.2004.08.006
3. Benner, P., Mehrmann, V., Sorensen, D.: Dimension Reduction of Large-Scale Systems. Lecture Notes in Computational Science and Engineering, vol. 45. Springer, Berlin (2005)
4. Bozzini, M., Lenarduzzi, L., Schaback, R.: Adaptive interpolation by scaled multiquadrics. Adv. Comput. Math. **16**(4), 375–387 (2002). doi:10.1023/A:1014584220418
5. Bozzini, M., Rossini, M., Schaback, R.: Generalized Whittle–Matérn and polyharmonic kernels. Adv. Comput. Math. (2012). doi:10.1007/s10444-012-9277-9
6. Buhmann, M.D.: Radial Basis Functions: Theory and Implementations. Cambridge University Press, Cambridge (2003)
7. Chaturantabut, S., Sorensen, D.C.: Nonlinear model reduction via discrete empirical interpolation. SIAM J. Sci. Comput. **32**(5), 2737–2764 (2010). doi:10.1137/090766498
8. Ehrhardt, K., Steinbach, M.C.: Nonlinear optimization in gas networks. ZIB Report ZR-03-46, Konrad-Zuse-Zentrum fuer Informationstechnik (2003)
9. Fasshauer, G.E.: Meshfree Approximation Methods with Matlab (with CD-ROM). World Scientific, Singapore (2007)
10. Fornberg, B., Wright, G.: Stable computation of multiquadric interpolants for all values of the shape parameter. Comput. Math. Appl. **48**(5–6), 853–867 (2004). doi:10.1016/j.camwa.2003.08.010
11. Fornberg, B., Zuev, J.: The Runge phenomenon and spatially variable shape parameters. Comput. Math. Appl., 379–398 (2006)
12. Fornberg, B., Larsson, E., Flyer, N.: Stable computations with Gaussian radial basis functions. SIAM J. Sci. Comput. **33**(2), 869–892 (2011). doi:10.1137/09076756X
13. Herty, M., Mohring, J., Sachers, V.: A new model for gas flow in pipe networks. Math. Methods Appl. Sci. **33**(7), 845–855 (2010). doi:10.1002/mma.1197
14. Jones, E., Oliphant, T., Peterson, P., et al.: SciPy: open source scientific tools for Python (2001). http://www.scipy.org/
15. Kansa, E., Carlson, R.: Improved accuracy of multiquadric interpolation using variable shape parameters. Comput. Math. Appl. **24**(12), 99–120 (1992). doi:10.1016/0898-1221(92)90174-G
16. LIWACOM Informationstechnik GmbH, Simone research group, Essen: Simone Software: Gleichungen und Methoden (2004)
17. Pazouki, M., Schaback, R.: Bases for kernel-based spaces. J. Comput. Appl. Math. **236**(4), 575–588 (2011). doi:10.1016/j.cam.2011.05.021
18. Schaback, R.: Approximation by radial basis functions with finitely many centers. Constr. Approx. **12**(3), 331–340 (1996). doi:10.1007/BF02433047
19. Schaback, R.: Native Hilbert spaces for radial basis functions I. In: New Developments in Approximation Theory. International Series of Numerical Mathematics, vol. 132, pp. 255–282. Birkhäuser, Basel (1997)

20. Scheuerer, M., Schaback, R., Schlather, M., Feld, I.N., et al.: Interpolation of spatial data—a stochastic or a deterministic problem. Preprint, Universität Göttingen (2011). http://num.math.uni-goettingen.de/schaback/research/papers/IoSD.pdf
21. Schilders, W., van der Vorst, H., Rommes, J.: Model Order Reduction: Theory, Research Aspects and Applications. Springer, Berlin (2008)
22. Sirovich, L.: Turbulence and the dynamics of coherent structures. Parts I–III. Q. Appl. Math. **45**(3), 561–590 (1987)
23. Steinbach, M.C.: On PDE solution in transient optimization of gas networks. J. Comput. Appl. Math. **203**(2), 345–361 (2007). doi:10.1016/j.cam.2006.04.018
24. Wendland, H.: Scattered Data Approximation. Cambridge University Press, Cambridge (2010)

Aerodynamic Shape Optimization by Space Mapping

Leifur Leifsson, Slawomir Koziel, Eirikur Jonsson, and Stanislav Ogurtsov

Abstract Space mapping (SM) has been successfully applied in various fields of engineering and science. However, despite its potential, SM has only recently been applied in aerospace engineering. This chapter describes recent advances in aerodynamic design and optimization using SM. In particular, a detailed formulation of the optimization methodology is provided, as well as several applications involving the design of transonic airfoils and wings.

Keywords Aerodynamic shape optimization · Space mapping · CFD

1 Introduction

Aerodynamic shape optimization (ASO) is a field of engineering that couples optimization methods and fluid flow analysis to design streamlined surfaces. ASO is important in the design of a number of engineering systems and devices, such as aircraft [1], turbomachinery [2], and automotive [3] and marine vehicles [4]. Nowadays, high-fidelity computational fluid dynamic (CFD) simulations are widely used in the design process. Although CFD analyses are accurate and reliable, they can be computationally expensive. Therefore, efficient and robust design algorithms are essential for rapid optimization.

Computationally efficient ASO can be performed using surrogate-based optimization (SBO) techniques [5–7]. In SBO, direct optimization of the high-fidelity

L. Leifsson (✉) · S. Koziel · E. Jonsson · S. Ogurtsov
Engineering Optimization & Modeling Center, School of Science and Engineering, Reykjavik University, Menntavegur 1, 101 Reykjavik, Iceland
e-mail: leifurth@ru.is

S. Koziel
e-mail: koziel@ru.is

E. Jonsson
e-mail: eirikurjon07@ru.is

S. Ogurtsov
e-mail: stanislav@ru.is

S. Koziel, L. Leifsson (eds.), *Surrogate-Based Modeling and Optimization*,
DOI 10.1007/978-1-4614-7551-4_10,

CFD model is replaced by an iterative correction-prediction process where a surrogate model (a computationally cheap representation of the high-fidelity one) is constructed and subsequently exploited to obtain an approximate location of the high-fidelity model optimal design.

A surrogate model can be constructed by approximating sampled high-fidelity model data using, e.g., polynomial approximation [5], radial basis functions [6, 8], kriging [7, 9–11], neural networks [12, 13], or support vector regression [14] (response surface approximation surrogates) or by correcting/enhancing a physics-based low-fidelity model (physical surrogates) [7, 15]. Approximation surrogates usually require a large number of high-fidelity model evaluations to ensure decent accuracy, and the number of samples typically grows exponentially with the number of design variables. On the other hand, approximation surrogates can be a basis of efficient global optimization techniques [6]. Various techniques of updating the training data set (the so-called infill criteria [6]) have been developed that aim at obtaining global modeling accuracy, locating a globally optimal design, or achieving a trade-off between the two, particularly in the context of kriging interpolation [6].

Physics-based surrogate models are not as versatile as approximation ones, because they rely on the underlying low-fidelity model (a simplified description of the system). Low-fidelity models can be obtained by neglecting certain second-order effects, using simplified equations, or—probably the most versatile approach—by exploiting the same CFD solver as used to evaluate the high-fidelity model but with a coarser mesh and/or relaxed convergence criteria [16]. It seems that physical surrogates have the potential to offer better efficiency in terms of reducing the computational cost of the design process. The reason is that the knowledge about the system of interest embedded into the low-fidelity model allows us to construct a quality surrogate model using a limited amount of high-fidelity model data. For many practical algorithms, only a single high-fidelity model evaluation is sufficient [7, 17]. For the same reason, physical surrogates have much better generalization capability than the approximation models [17].

Several SBO algorithms exploiting physical surrogates have been proposed in the literature, including approximation and model management optimization (AMMO) [18], space mapping (SM) [17, 19], manifold mapping (MM) [20], and, recently, shape-preserving response prediction (SPRP) [21]. All of these methods differ in the specific method of using the low-fidelity model to create the surrogate. Space mapping is probably the most popular approach of this kind. It was originally developed for simulation-driven design in microwave engineering [17]; however, it is currently becoming more and more popular in other areas of engineering and science (see [17, 19]). Despite its potential, space mapping has only recently been applied in ASO [22–24].

In this chapter, we describe a computationally efficient ASO methodology which employs physics-based surrogate models created by space mapping [23, 24]. Section 2 briefly describes aerodynamic shape design and the optimization problem. The optimization methodology is described in detail in Sect. 3. Applications of the method to transonic airfoil and wing design are given in Sects. 4, 5. Section 6 summarizes the chapter.

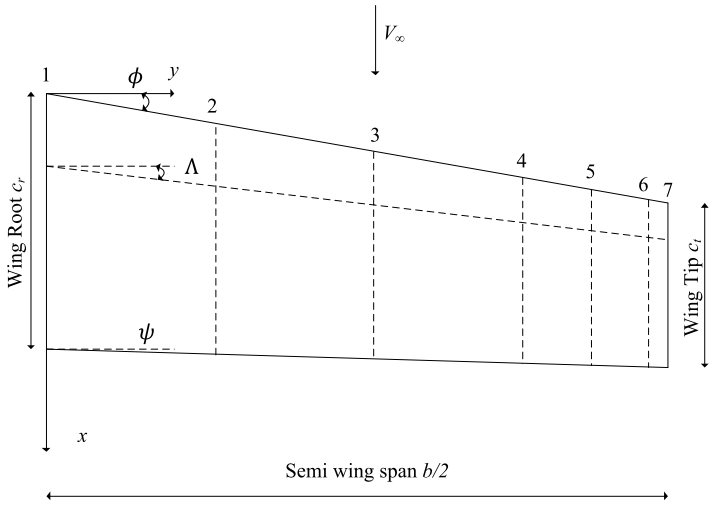


Fig. 1 Planform view of a trapezoidal wing (or a turbine blade) of a semi-span $b/2$ and quarter chord sweep angle Λ. Span stations are marked 1 through 7 and the free-stream velocity is V_∞. Leading edge and trailing edge angles ϕ and ψ are also shown. Other design parameters are not shown

2 Aerodynamic Shape Design

The fundamentals of aerodynamic shape design are briefly addressed in this section. In particular, the geometry description, figures of merit, and the optimization problem are defined.

2.1 Geometry

A three-dimensional streamlined aerodynamic surface is depicted in Fig. 1, which could, for example, represent a simple aircraft wing or a turbine blade. Design parameters controlling the planform shape include the semi-span ($b/2$), the quarter chord wing sweep angle (Λ), the thickness-to-chord ratio (t/c) at each span station, the wing taper ratio (λ), and the twist distribution (γ).

At each span station (numbered 1 through 7) of the surface, the cross section is defined by an airfoil profile such as the one in Fig. 2. The number of span stations can be larger or fewer than shown here and depends on the particular design scenario. A straight line wrap is often assumed between the span stations. The airfoil shapes are characterized by the chord length (c), thickness distribution (t), and camber distribution. Designable parameters controlling the overall shape depend on the parameterization technique employed by the designer. Examples of such techniques include the National Advisory Committee for Aeronautics (NACA) four-digit airfoil shapes [25], the Hicks and Henne bump functions [26], and the Bezier-PARSEC method [27], each with different types and numbers of control parameters.

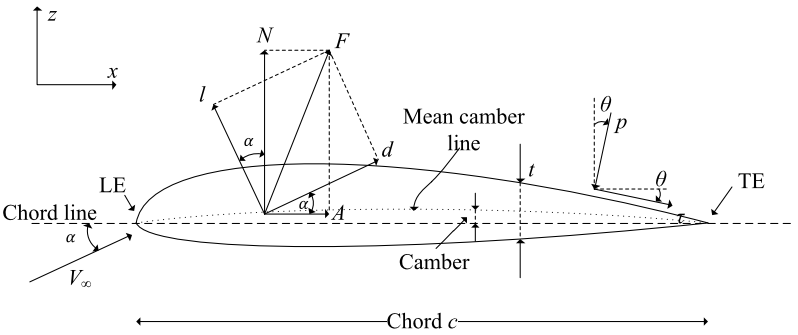


Fig. 2 Airfoil wing cross section (*solid line*) of thickness t and chord length c. V_∞ is the free-stream velocity and is at an angle of attack α relative to the x-axis. F is the resulting aerodynamic force, where l is the lift force, perpendicular to V_∞, and d is the drag force, parallel to V_∞. p is the pressure acting normal to a surface element ds. τ is the viscous wall shear stress acting parallel to the surface element. θ is the angle that p and τ make relative to the z-axis and x-axis, respectively, where a positive angle is clockwise

2.2 Figures of Merit

In general, a streamlined aerodynamic surface is designed to provide a certain lift at a minimum drag. In ASO, the typical figures of merit are the lift and drag coefficients. The lift coefficient (for a three-dimensional surface) is defined as

$$C_L = \frac{L}{q_\infty S}, \tag{1}$$

and the drag coefficient as

$$C_D = \frac{D}{q_\infty S}, \tag{2}$$

where S is a reference area (usually chosen as the planform area), L and D are the magnitude of the total lift and drag forces, respectively, and the dynamic pressure q_∞ is defined as

$$q_\infty = \frac{1}{2}\rho_\infty V_\infty, \tag{3}$$

where ρ_∞ is the free-stream density and V_∞ is the magnitude of the free-stream velocity.

The forces acting on the surface are calculated from the results of a numerical simulation of the flow past it. In particular, the lift and drag coefficients can be calculated as

$$C_L = -C_A \sin\alpha + C_N \cos\alpha, \tag{4}$$

and

$$C_D = C_A \cos\alpha + C_N \sin\alpha, \tag{5}$$

Table 1 Various problem formulations for aerodynamic shape optimization. Typically, constraints on the minimum allowable cross-sectional area are also included, i.e., $c_2(\mathbf{x}) = A_{\min}(\mathbf{x}) - A(\mathbf{x}) \leq 0$, where $A_{\min}(\mathbf{x})$ is the minimum cross-sectional area at a given span station. $C_{p.t}$ is a target pressure distribution

Case	$f(\mathbf{x})$	$c_1(\mathbf{x})$
Lift maximization	$-C_L(\mathbf{x})$	$C_D(\mathbf{x}) - C_{D.\min} \leq 0$
Drag minimization	$C_D(\mathbf{x})$	$C_{L.\max} - C_L(\mathbf{x}) \leq 0$
L/D maximization	$-C_L(\mathbf{x})/C_D(\mathbf{x})$	$C_{L.\max} - C_L(\mathbf{x}) \leq 0$
Inverse design	$1/2 \int (C_p(\mathbf{x}) - C_{p.t})^2 \, ds$	

respectively. Here, the nondimensional force coefficients parallel to the x- and z-axes, C_A and C_N, respectively, are calculated by integrating the pressure distribution (C_p) and the skin friction distribution (C_f) over the surface as

$$C_A = \oint (C_p \sin\theta + C_f \cos\theta) \, ds, \tag{6}$$

and

$$C_N = \oint (-C_p \cos\theta + C_f \sin\theta) \, ds, \tag{7}$$

where ds is the length of the surface panel element and θ is the angle of the panel relative to the x-axis (see definition in Fig. 2).

2.3 Optimization Problem

Generally, aerodynamic shape optimization can be formulated as a nonlinear minimization problem; i.e., for a given operating condition, solve

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & c_j(\mathbf{x}) \leq 0 \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \end{aligned} \tag{8}$$

where $f(\mathbf{x})$ is the objective function, $\mathbf{x}$ is the design variable vector, $c_j(\mathbf{x})$ is a design constraint ($j = 1, \ldots, N$ and N is the number of constraints), and $\mathbf{l}$ and $\mathbf{u}$ are the lower and upper bounds for the design variables, respectively. The design variables and the detailed formulation are problem-specific, but typical formulations are shown in Table 1. Other constraints such as mathematical models describing the structural weight of the wing are often included in optimization [28].

3 Space Mapping for Aerodynamic Design

In this section, we describe a space mapping (SM) optimization methodology for aerodynamic shape design. First, the aerodynamic optimization problem described in the previous section is formulated for the SM approach. Then, we briefly recall some SM basics, and define the surrogate model construction. Finally, the SM algorithm is described.

3.1 Problem Formulation

A simulation-driven design can be formulated as a nonlinear minimization problem as noted before. Refining (8), we define

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} H\big(f(\mathbf{x})\big), \tag{9}$$

where $\mathbf{x}$ is a vector of design parameters, f the high-fidelity model to be minimized at $\mathbf{x}$, and H is the objective function. $\mathbf{x}^*$ is the optimum design vector. The high-fidelity model will represent aerodynamic forces, the lift and drag coefficients, as well as other scalar responses such as the cross-sectional area A of the wing at the location of interest. The area response can be of a vector form $\mathbf{A}$ if one requires multiple area cross-sectional constraints at various locations on the wing, e.g., the wing root and the wing tip. The response will have the form

$$f(\mathbf{x}) = \begin{bmatrix} C_{L.f}(\mathbf{x}) & C_{D.f}(\mathbf{x}) & A_f(\mathbf{x}) \end{bmatrix}^T, \tag{10}$$

where $C_{L,f}$ and $C_{D,f}$ are the lift and drag coefficients, respectively, generated by the high-fidelity model. We are interested in maximizing lift or minimizing drag, so the objective function will take the form

$$H\big(f(\mathbf{x})\big) = -C_L, \tag{11}$$

or

$$H\big(f(\mathbf{x})\big) = C_D, \tag{12}$$

respectively, with the design constraints denoted as

$$C\big(f(\mathbf{x})\big) = \begin{bmatrix} c_1(f(\mathbf{x})) & \ldots & c_k(f(\mathbf{x})) \end{bmatrix}^T. \tag{13}$$

Maximizing lift will yield two nonlinear design constraints for drag and area,

$$c_1\big(f(\mathbf{x})\big) = C_{D,f}(\mathbf{x}) - C_{D,\max} \leq 0, \tag{14}$$

$$c_2\big(f(\mathbf{x})\big) = -A_f(\mathbf{x}) + A_{\min} \leq 0. \tag{15}$$

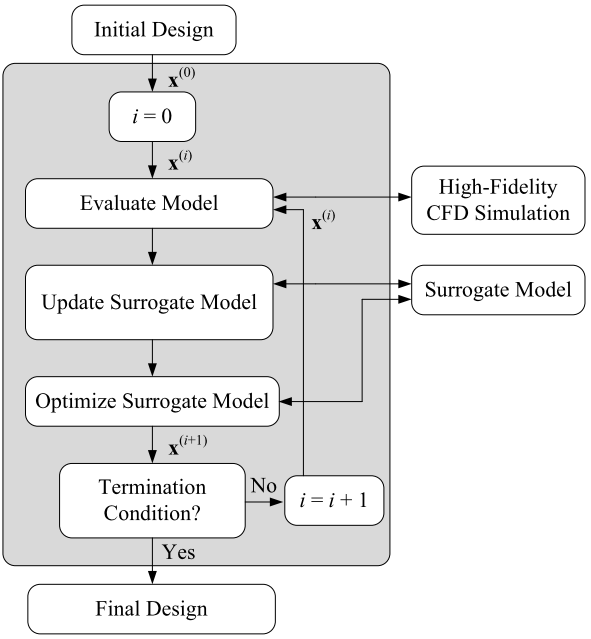


Fig. 3 A flow diagram of a generic SBO algorithm

Similarly, minimizing drag we have two nonlinear design constraints for lift and area,

$$c_1\big(f(\mathbf{x})\big) = -C_{L,f}(\mathbf{x}) + C_{L,\min} \leq 0, \tag{16}$$

$$c_2\big(f(\mathbf{x})\big) = -A_f(\mathbf{x}) + A_{\min} \leq 0. \tag{17}$$

3.2 Space Mapping Basics

Space mapping (SM) [17, 19] is a surrogate-based optimization (SBO) technique where the computational burden is shifted from an expensive high-fidelity model (or fine model), denoted by f, to a cheaper model, the surrogate, denoted by s, where the surrogate is iteratively optimized and updated. The flow of a typical SBO algorithm is shown in Fig. 3.

Starting from a initial design $\mathbf{x}^{(0)}$, the genetic SM algorithm produces a sequence $\mathbf{x}^{(i)}$, $i = 0, 1, \ldots$ of an approximate solution to (9) as

$$\mathbf{x}^{(i+1)} = \arg\min_{\mathbf{x}} H\big(s^{(i)}(\mathbf{x})\big), \tag{18}$$

where

$$s^{(i)}(\mathbf{x}) = \big[C^{(i)}_{L.s}(\mathbf{x}) \quad C^{(i)}_{D.s}(\mathbf{x}) \quad A_s(\mathbf{x})^{(i)}\big]^T \tag{19}$$

is the surrogate model at iteration i. As previously described, the accurate high-fidelity CFD model f is accurate but computationally expensive. Using SM, the surrogate s is a composition of the low-fidelity CFD model c and a simple linear transformation to correct the low-fidelity model response [17]. The corrected response is denoted as $s(\mathbf{x}, \mathbf{p})$, where $\mathbf{p}$ represents a set of model parameters and at iteration i the surrogate is

$$s^{(i)}(\mathbf{x}) = s(\mathbf{x}, \mathbf{p}). \tag{20}$$

The SM parameters $\mathbf{p}$ are determined through a parameter extraction (PE) process [17]. In general, this process is a nonlinear optimization problem, where the objective is to minimize the misalignment of the surrogate response at some or all previous iterations of the high-fidelity model data points [17]. The PE optimization problem can be defined as

$$\mathbf{p}^{(i)} = \arg\min_{\mathbf{p}} \sum_{k=0}^{i} w_{i,k} \left\| f\left(\mathbf{x}^{(k)}\right) - s\left(\mathbf{x}^{(k)}, \mathbf{p}\right) \right\|^2, \tag{21}$$

where $w_{i,k}$ are weight factors that control how much impact previous iterations affect the SM parameters. Popular choices are

$$w_{i,k} = 1 \quad \forall i, k, \tag{22}$$

and

$$w_{i,k} = \begin{cases} 1 & k = i, \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

In the latter case, the parameters only depend on the most recent SM iteration.

Examples of SM surrogate models include input SM, where $s(\mathbf{x}, \mathbf{p}) = s(\mathbf{x}, \mathbf{q}) = c(\mathbf{x} + \mathbf{q})$ (parameter shift) or $s(\mathbf{x}, \mathbf{p}) = s(\mathbf{x}, \mathbf{B}, \mathbf{q}) = c(\mathbf{B}\mathbf{x} + \mathbf{q})$ (parameter shift and scaling), output SM, with $s(\mathbf{x}, \mathbf{p}) = s(\mathbf{x}, \mathbf{A}) = \mathbf{A}c(\mathbf{x})$ (multiplicative response correction) or $s(\mathbf{x}, \mathbf{p}) = s(\mathbf{x}, \mathbf{d}) = c(\mathbf{x}) + \mathbf{d}$ (additive response correction), and a few others such as implicit SM [29] and frequency SM [30].

3.3 Surrogate Model Construction

The SM surrogate model s is a composition of a low-fidelity model c and corrections or linear transformations, where the model parameters p are extracted using one of the PE processes described above. PE and surrogate optimization create a certain overhead on the whole process, and this overhead can be a significant part of the overall computational cost. This is mainly due to the fact that the physics-based low-fidelity models are, in general, relatively expensive to evaluate compared to the functional-based ones. Despite this, SM may be beneficial [31].

This problem can be alleviated by exploiting the output SM with both multiplicative and additive response corrections where the surrogate model parameters are extracted analytically. The surrogate model is then written as

$$s^{(i)}(\mathbf{x}) = \mathbf{A}^{(i)} \circ c(\mathbf{x}) + \mathbf{D}^{(i)} + \mathbf{q}^{(i)} \tag{24}$$

$$= \left[a_L^{(i)} C_{L.c}(\mathbf{x}) + d_L^{(i)} + q_L^{(i)} \quad a_D^{(i)} C_{D.c}(\mathbf{x}) + d_D^{(i)} + q_D^{(i)} \quad A_c(\mathbf{x})\right]^T, \tag{25}$$

where $\circ$ is a component-wise multiplication. No mapping is needed for the area $A_c(\mathbf{x})$, where $A_c(\mathbf{x}) = A_f(\mathbf{x})\ \forall \mathbf{x}$ since the low- and high-fidelity models represent the same geometry. Parameters $\mathbf{A}^{(i)}$ and $\mathbf{D}^{(i)}$ are obtained using

$$\left[\mathbf{A}^{(i)}, \mathbf{D}^{(i)}\right] = \arg\min_{\mathbf{A},\mathbf{D}} \sum_{k=0}^{i} \left\| f\left(\mathbf{x}^{(k)}\right) - \mathbf{A} \circ c\left(\mathbf{x}^{(k)}\right) + \mathbf{D} \right\|^2, \tag{26}$$

where $w_{i,k} = 1$; i.e., all previous iteration points are used to globally improve the response of the low-fidelity model. The additive term $q^{(i)}$ is defined so as to ensure this perfect match between the surrogate and the high-fidelity model at design $\mathbf{x}^{(i)}$, namely $f(\mathbf{x}^{(i)}) = s(\mathbf{x}^{(i)})$ or a zero-order consistency [18]. We can write the additive term as

$$q^{(i)} = f\left(\mathbf{x}^{(i)}\right) - \left[\mathbf{A}^{(i)} \circ c\left(\mathbf{x}^{(i)}\right) + \mathbf{D}^{(i)}\right]. \tag{27}$$

Since an analytical solution exists for $\mathbf{A}^{(i)}, \mathbf{D}^{(i)}$, and $\mathbf{q}^{(i)}$, there is no need for nonlinear optimization solving (21) to obtain parameters. We can obtain $\mathbf{A}^{(i)}$ and $\mathbf{D}^{(i)}$ as

$$\begin{bmatrix} a_L^{(i)} \\ d_L^{(i)} \end{bmatrix} = \left(\mathbf{C}_L^T \mathbf{C}_L\right)^{-1} \mathbf{C}_L^T \mathbf{F}_L, \tag{28}$$

$$\begin{bmatrix} a_D^{(i)} \\ d_D^{(i)} \end{bmatrix} = \left(\mathbf{C}_D^T \mathbf{C}_D\right)^{-1} \mathbf{C}_D^T \mathbf{F}_D, \tag{29}$$

where

$$\mathbf{C}_L = \begin{bmatrix} C_{L.c}(\mathbf{x}^{(0)}) & C_{L.c}(\mathbf{x}^{(1)}) & \dots & C_{L.c}(\mathbf{x}^{(i)}) \\ 1 & 1 & \dots & 1 \end{bmatrix}^T, \tag{30}$$

$$\mathbf{F}_L = \begin{bmatrix} C_{L.f}(\mathbf{x}^{(0)}) & C_{L.f}(\mathbf{x}^{(1)}) & \dots & C_{L.f}(\mathbf{x}^{(i)}) \\ 1 & 1 & \dots & 1 \end{bmatrix}^T, \tag{31}$$

$$\mathbf{C}_D = \begin{bmatrix} C_{D.c}(\mathbf{x}^{(0)}) & C_{D.c}(\mathbf{x}^{(1)}) & \dots & C_{D.c}(\mathbf{x}^{(i)}) \\ 1 & 1 & \dots & 1 \end{bmatrix}^T, \tag{32}$$

$$\mathbf{F}_D = \begin{bmatrix} C_{D.f}(\mathbf{x}^{(0)}) & C_{D.f}(\mathbf{x}^{(1)}) & \dots & C_{D.f}(\mathbf{x}^{(i)}) \\ 1 & 1 & \dots & 1 \end{bmatrix}^T, \tag{33}$$

and these are the least-square optimal solutions to the linear regression problems

$$\mathbf{C}_L a_L^{(i)} + d_L^{(i)} = \mathbf{F}_L, \tag{34}$$

$$\mathbf{C}_D a_D^{(i)} + d_D^{(i)} = \mathbf{F}_D. \tag{35}$$

Note that $\mathbf{C}_L^T \mathbf{C}_L$ and $\mathbf{C}_L^T \mathbf{C}_L$ are nonsingular for $i > 1$ and $\mathbf{x}^{(k)} \neq \mathbf{x}^{(i)}$ for $k \neq i$. For $i = 1$ only the multiplicative SM correction with $\mathbf{A}^{(i)}$ is used.

3.4 Algorithm

The SM optimization algorithm, exploiting the trust region convergence safeguard [6], is as follows:

1. Set $i = 0$; Select λ, the trust region radius; Evaluate the high-fidelity model at the initial solution, $f(\mathbf{x}^{(0)})$;
2. Using data from the low-fidelity model c, and f at $\mathbf{x}^{(k)}, k = 0, 1, \ldots, i$, set up the SM surrogate $s^{(i)}$; Perform PE;
3. Optimize $s^{(i)}$ to obtain $\mathbf{x}^{(i+1)}$;
4. Evaluate $f(\mathbf{x}^{(i+1)})$;
5. If $H(f(\mathbf{x}^{(i+1)})) < H(f(\mathbf{x}^{(i)}))$, accept $\mathbf{x}^{(i+1)}$; Otherwise set $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)}$;
6. Update λ;
7. Set $i = i + 1$;
8. If the termination condition is not satisfied, go to 2, else proceed;
9. End; Return $\mathbf{x}^{(i)}$ as the optimum solution.

Note that the trust region parameter λ is updated after each iteration.

4 Airfoil Design

The use of the SM algorithm is demonstrated here for the design optimization of airfoil shapes at transonic flow conditions. A steady inviscid Euler CFD model is used with a structured grid for both the high- and low-fidelity models, but with different grid resolution and convergence criteria (variable-resolution models). The algorithm is applied to both lift maximization and drag minimization.

4.1 High-Fidelity Model

The flow is assumed to be steady, inviscid, and adiabatic with no body forces. The compressible Euler equations are taken to be the governing fluid flow equations (see, e.g., [32]). The solution domain boundaries are placed at 25 chord lengths in front of the airfoil, 50 chord lengths behind it, and 25 chord lengths above and below

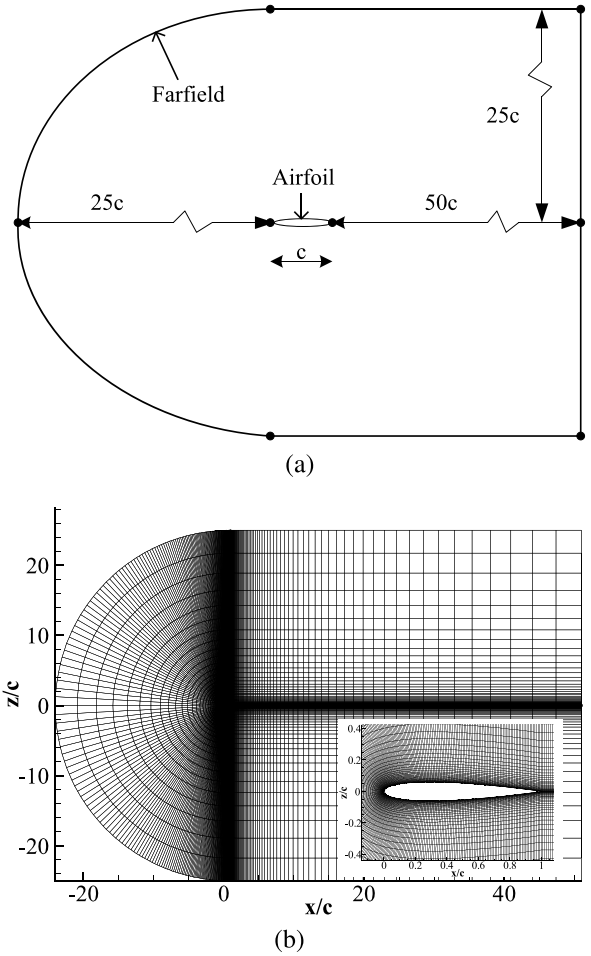
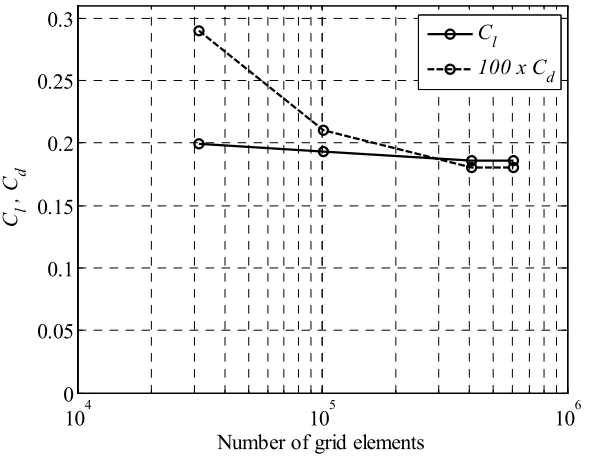


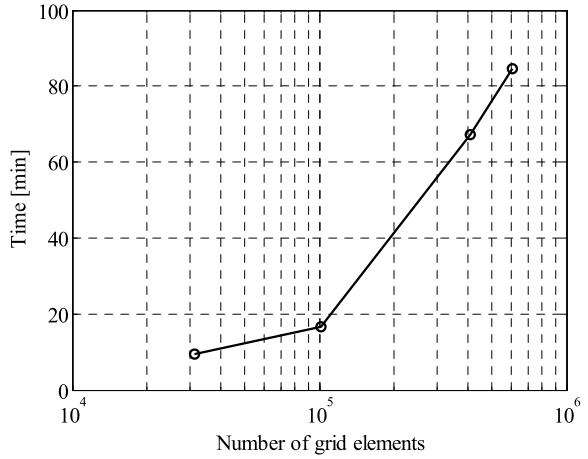
Fig. 4 Particulars of the two-dimensional solution domain and the grid: (**a**) a sketch of the computational domain for the flow past an airfoil with a chord length c, (**b**) example grid for the NACA 0012 airfoil

it (see Fig. 4). The computational meshes are of structured curvilinear body-fitted C-topology with elements clustering around the airfoil and growing in size with distance from the airfoil surface. The computer code ICEM CFD [33] is used for the mesh generation. The free-stream Mach number, static pressure, and angle of attack are prescribed at the far-field boundary.

Numerical fluid flow simulations are performed using the computer code FLUENT [34]. The flow solver is of an implicit density-based formulation, and the inviscid fluxes are calculated by an upwind-biased second-order spatially accurate Roe flux scheme. Asymptotic convergence to a steady state solution is obtained for each case. The iterative convergence of each solution is examined by monitoring the



(a)



(b)

Fig. 5 Grid convergence study using the NACA 0012 airfoil at Mach number $M_\infty = 0.75$ and angle of attack $\alpha = 1$ deg: (**a**) lift and drag coefficients versus the number of grid elements, and (**b**) the simulation time versus the number of grid elements

overall residual, which is the sum (over all the cells in the computational domain) of the L^2 norm of all the governing equations solved in each cell. In addition to this, the lift and drag forces (defined in Sect. 2) are monitored for convergence. The solution convergence criterion for the high-fidelity model is the one that occurs first of the following: a maximum residual of 10^{-6}, or a maximum number of iterations of 1,000.

A grid convergence study was performed using the NACA 0012 airfoil at Mach number $M_\infty = 0.75$ and angle of attack $\alpha = 1$ deg. The study, shown in Fig. 5a,

revealed that 407,676 mesh cells are needed for mesh convergence; thus, that particular mesh was used for the high-fidelity model. The overall simulation time for the case considered is around 67 min (Fig. 5b). The flow solver reached a converged solution after 352 iterations. The other meshes required around 350 to 500 iterations to converge, except the coarsest mesh, which terminated after 1,000 iterations, with an overall simulation time of around 9.5 min.

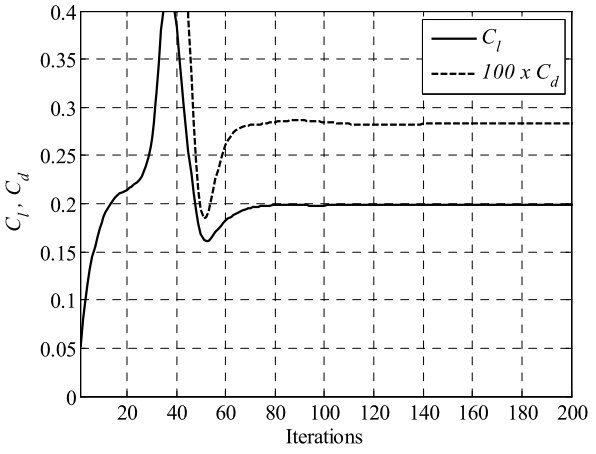
4.2 Low-Fidelity Model

The low-fidelity CFD model is constructed in the same way as the high-fidelity model, but with a coarser computational mesh and relaxed convergence criteria. For the low-fidelity model, we use the coarse mesh in the grid study presented in Fig. 5a, with 31,356 mesh cells. The flow solution history for the low-fidelity model, shown in Fig. 6a, indicates that the lift and drag coefficients are nearly converged after 80 to 100 iterations. The maximum number of iterations is set to 100 for the low-fidelity model. This reduced the overall simulation time to 1.5 min. A comparison of the pressure distributions, shown in Fig. 6b, indicates that the low-fidelity model, despite being based on a much coarser mesh and reduced flow solver iterations, captures the main features of the high-fidelity model pressure distribution quite well. The biggest discrepancy in the distributions is around the shock on the upper surface, leading to an overestimation of both lift and drag (Fig. 5a).

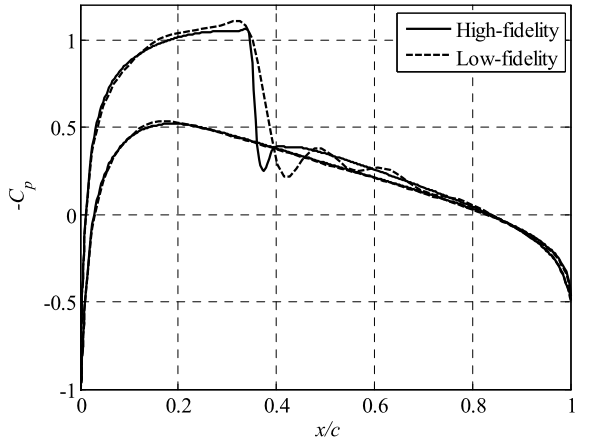
The ratio of simulation times of the high- and low-fidelity model in this particular case study is 43.8. In many cases, the solver does not fully converge with respect to the residuals and goes on up to 1,000 iterations. Then, the overall evaluation time of the high-fidelity model goes up to 170 min. In those cases, the ratio of simulation times of the high- and low-fidelity models is around 110. For simplicity, we will use a fixed value of 80 when estimating the equivalent number of function calls, i.e., when the number of low-fidelity function calls is added to the number of high-fidelity function calls.

4.3 Surrogate Model

The surrogate model is constructed using (24). The responses for a few randomly selected airfoil geometries using the low-fidelity model $c(\mathbf{x})$ (selected in the previous section) and the high-fidelity model $f(\mathbf{x})$ are shown in Fig. 7, as well as the globally corrected surrogates $A \circ c(\mathbf{x}) + D$, calculated using (26). Note that output SM is capable of substantially reducing the misalignment between the surrogate and high-fidelity model responses. The supplemental additive output SM term $q^{(i)}$ is only applied locally, as in (27), to further improve the surrogate model accuracy in the vicinity of the current design.



(a)



(b)

Fig. 6 Simulation results for NACA 0012 at Mach number $M_\infty = 0.75$ and angle of attack $\alpha = 1$ deg: (**a**) evolution of the lift and drag coefficients obtained by the low-fidelity model, (**b**) comparison of the pressure distributions obtained by the high- and low-fidelity models

4.4 Case Description

Four design cases are presented: three lift maximizations (Cases 1–3) and one drag minimization (Case 4). For Cases 1 through 3, the objective is to maximize the lift coefficient $C_{l.f}$, subject to constraints on drag ($C_{d.f} \leq C_{d.\max}$) and nondimensionalized airfoil cross-sectional area with the chord squared ($A \geq A_{\min}$). For Case 4, the objective is to minimize the drag, subject to constraints on lift coefficient ($C_{l.f} \geq C_{l.\min}$) and nondimensionalized airfoil cross-sectional area with the chord squared ($A \geq A_{\min}$).

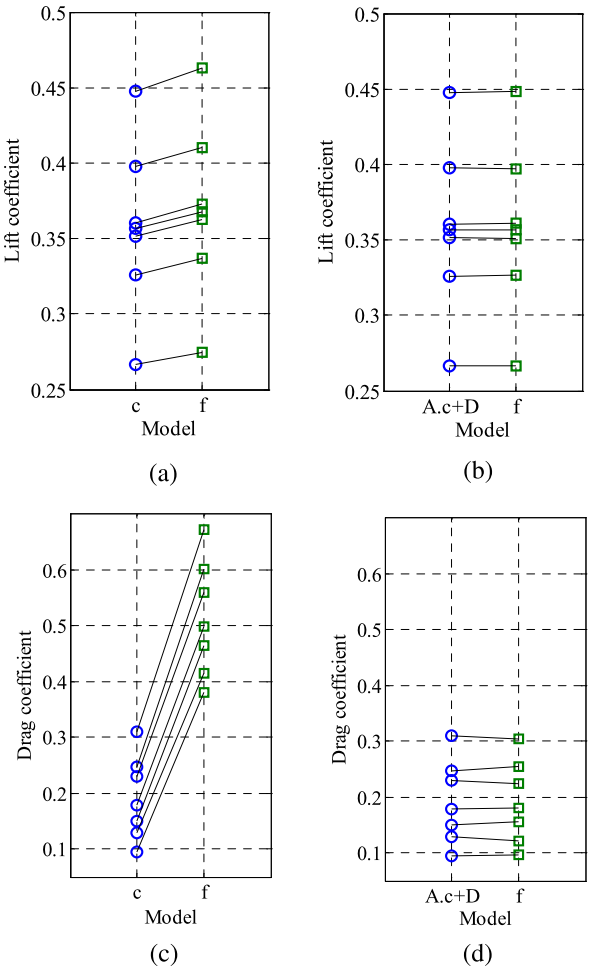


Fig. 7 The effect of output SM (26) on the low- and high-fidelity model alignment: (**a**) high- (∘) and low-fidelity (□) lift coefficient for selected airfoil geometries, (**b**) high- (∘) and output-S-M-aligned low-fidelity (□) lift coefficient for corresponding geometries, (**c**) the same as (**a**) but for the drag coefficient, (**d**) the same as (**b**) but for the drag coefficient. The model misalignment is greatly reduced by applying both multiplicative and additive output SM. The alignment is further improved by the supplemental local additive output SM term $q^{(i)}$ (cf. (27))

For simplicity, and because we have a small number of design variables, the NACA four-digit method is used to parameterize the airfoil shapes. Although these airfoils are intended for subsonic flow, it is used here for demonstration purposes only, as the method has only three design parameters. The particulars of the method are given in the Appendix.

The NACA four-digit airfoil design variables are m (the maximum ordinate of the mean camber line as a fraction of the chord), p (the chordwise position of

the maximum ordinate), and t/c (the thickness-to-chord ratio). The design vector can be written as $\mathbf{x} = [m\ p\ t/c]^T$. The side constraints on the design variables are $0 \leq m \leq 0.1$, $0.2 \leq p \leq 0.8$, and $0.05 \leq t/c \leq 0.2$. Details of the test cases and optimization results are given in Table 2. The results are compared with the direct design optimization of the high-fidelity model using the pattern search algorithm. For the surrogate model optimization in the SM algorithm, the pattern search algorithm is also used. The termination condition for the SM algorithm is $\|\mathbf{x}^{(i)} - \mathbf{x}^{(i-1)}\| < 10^{-3}$.

4.5 Results

Consider Case 1 in Table 2, where the initial airfoil design is NACA 2412 and the drag constraint is violated. The direct method and the SM algorithm obtain comparable optimized designs by reducing camber, placing the location of the maximum camber relatively aft, and reducing the thickness. The SM algorithm required 210 surrogate model evaluations (N_c) and 4 high-fidelity model evaluations (N_f), yielding an equivalent number of high-fidelity model evaluations of less than 7. The direct method required 96 high-fidelity model evaluations.

In Case 2, the initial design is NACA 2412, which is feasible for the assumed constraints. The SM algorithm is able to obtain a better optimized design than the direct method. The shape changes are similar to those of Case 1, except that the camber is increased. The effects on the pressure distribution can be observed in Fig. 8: the shock strength is reduced by reducing the thickness, and the aft camber location opens up the pressure distribution behind the shock to increase the lift. The SM algorithm required less than 9 equivalent high-fidelity model evaluations (260 surrogate and 5 high-fidelity).

The optimization history for Case 2 is shown in Fig. 9. In particular, one can observe a convergence plot, as well as the evolution of the objective function, the lift coefficient, and the drag coefficient. It follows that the SM algorithm exhibits a good convergence pattern and enforces the drag limitation to be satisfied while increasing the lift coefficient as much as possible.

Case 3 has an initial design with higher camber and thinner section than the other cases, namely the NACA 3210. The SM algorithm achieves a better design than the direct method. Now the camber is reduced, but the location of the maximum camber is again moved aft and the thickness is slightly reduced. Less than 9 equivalent high-fidelity model evaluations are required.

In Case 4, the drag minimization case, the initial design is NACA 2412 and the lift constraint is slightly violated. Similar optimized designs are obtained by the direct method and the SM algorithm. The camber is reduced, the maximum camber moved aft, and the thickness reduced. As a result, the shock is weakened, and the lift improved by opening the pressure distribution behind the shock (see Fig. 10). The SM algorithm required 5 equivalent high-fidelity model evaluations (160 surrogate and 3 high-fidelity), whereas the direct method required 110.

Overall, it can be observed that the SM performance is consistent across the considered test cases. The average airfoil design cost is equivalent to about 5 to 9

Table 2 Numerical results for four test cases. Shown are results for the initial design, direct optimization, and optimization using space mapping

Variable	Case 1 (Lift maximization) $M_\infty = 0.75, \alpha = 0°$, $C_{d,\max} = 0.005$, $A_{\min} = 0.075$			Case 2 (Lift maximization) $M_\infty = 0.70, \alpha = 1°$, $C_{d,\max} = 0.006$, $A_{\min} = 0.075$		
	Initial	Direct	SM	Initial	Direct	SM
m	0.02	0.0140	0.0150	0.02	0.0200	0.0237
p	0.40	0.7704	0.7463	0.40	0.7467	0.6531
t/c	0.12	0.1150	0.1140	0.12	0.1200	0.1148
C_l	0.4745	0.5572	0.5650	0.5963	0.8499	0.8909
C_{dw}	0.0115	0.0050	0.0050	0.0047	0.0060	0.0060
A	0.0808	0.0774	0.0767	0.0808	0.0808	0.0773
N_c	–	0	210	–	0	260
N_f	–	96	4	–	59	5
Cost	–	96	<7	–	59	<9

Variable	Case 3 (Lift maximization) $M_\infty = 0.75, \alpha = 1°$, $C_{d,\max} = 0.0041$, $A_{\min} = 0.065$			Case 4 (Drag minimization) $M_\infty = 0.70, \alpha = 1°$, $C_{l,\min} = 0.6$, $A_{\min} = 0.075$		
	Initial	Direct	SM	Initial	Direct	SM
m	0.03	0.0080	0.0100	0.02	0.0180	0.0180
p	0.20	0.6859	0.6929	0.40	0.5207	0.5290
t/c	0.10	0.1044	0.0980	0.12	0.1141	0.1113
C_l	0.8035	0.4641	0.5281	0.5963	0.6001	0.6002
C_{dw}	0.0410	0.0041	0.0041	0.0047	0.0019	0.0017
A	0.0675	0.0703	0.0659	0.0808	0.0768	0.0749
N_c	–	0	260	–	0	160
N_f	–	121	5	–	110	3
Cost	–	121	<9	–	110	5

high-fidelity model evaluations, which corresponds to a cost savings of 80 percent or more depending on a test case, when compared to direct airfoil optimization using pattern search.

5 Wing Design

In this section, the SM algorithm is demonstrated for the design of a three-dimensional wing shape. Again, transonic flow is considered, but a steady viscous Reynolds-averaged Navier–Stokes (RANS) CFD model is used with an unstruc-

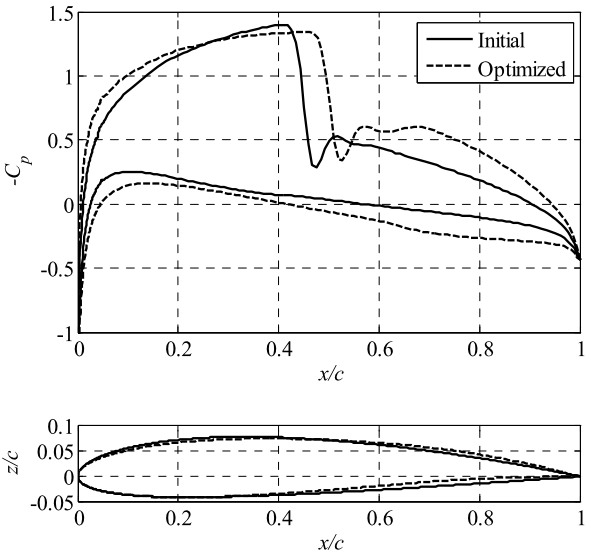


Fig. 8 Initial and optimized pressure distributions and airfoil shapes for Case 2

tured grid. Still, the variable-resolution approach is employed. One design case is considered, but optimization runs for two different initial designs are presented.

5.1 High-Fidelity Model

The flow is steady, compressible, viscous, and without body forces, mass diffusion, chemical reactions, or external heat addition. The RANS equations with the one-equation Spalart-Allmaras turbulence model [32] are solved. Air is modeled by the ideal gas law and the Sutherland dynamic viscosity model.

The far field is configured in a box topology where the wing root airfoil is placed in the center of the symmetry plane, with its leading edge placed at the origin. The far field extends 100 chord lengths in all directions upstream, above, below, and aft of the wing. The computational domain, along with the boundary conditions, is shown in Fig. 11.

An unstructured tri/tetra shell grid is created on all surfaces. The shell grid from the wing is then extruded into the volume where the volume is flooded with tri/tetra elements. The grid is made dense close to the wing, where it then gradually grows in size as it moves away from the wing surfaces. To capture the viscous boundary layer an inflation layer or a prism layer is created on the wing surfaces as well. The grid is generated using ANSYS ICEM CFD and is shown in Fig. 12.

In the stream-wise direction, the number of elements on the wing is set to 100 on both the upper and lower surfaces. The bigeometric bunching law with a growth ratio of 1.2 is employed in the stream-wise direction over the wing to obtain a more dense

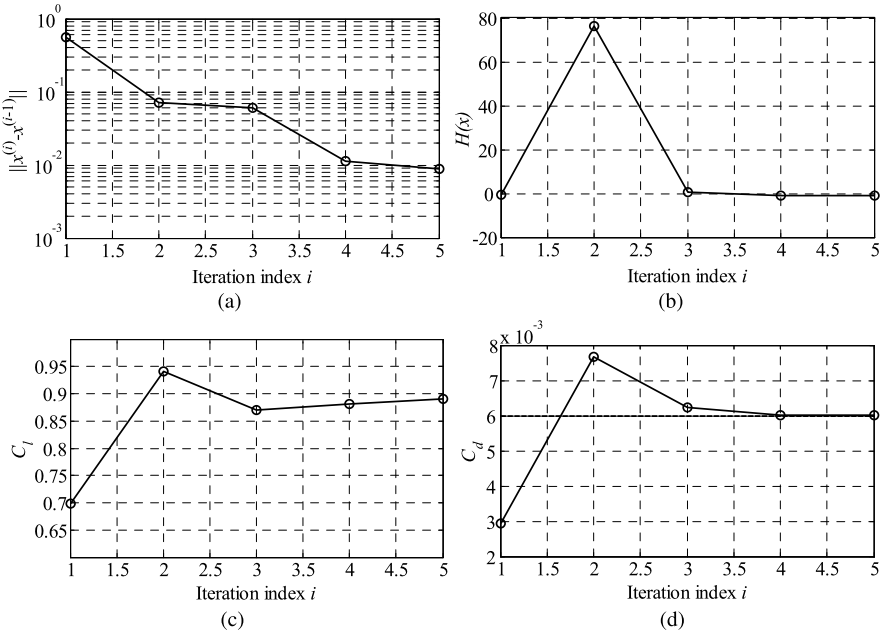


Fig. 9 Optimization history for Case 2: (**a**) convergence plot, (**b**) evolution of the objective function, (**c**) evolution of the lift coefficient, and (**d**) evolution of the drag coefficient (drag constraint marked using a *dashed horizontal line*). The graphs show all high-fidelity function evaluations performed in the optimization

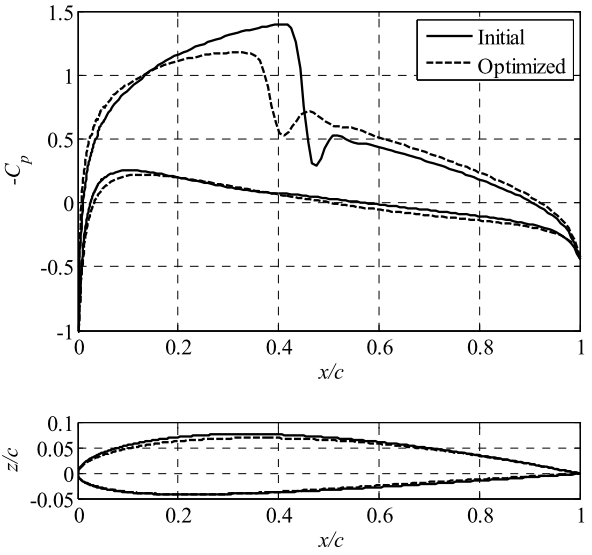


Fig. 10 Initial and optimized pressure distributions and airfoil shapes for Case 4

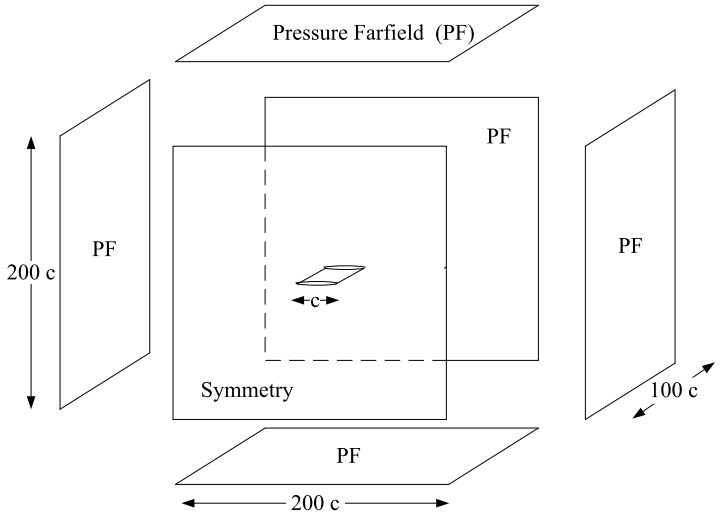


Fig. 11 Far field configured as a box topology. The leading edge of the wing root airfoil is placed at $(x, y, z) = (0, 0, 0)$

element distribution at the leading edge and the trailing edge. This is done in order to capture the high pressure gradient at the leading edge and potential separation at the trailing edge. The minimum element size of the wing in the stream-wise direction is set to 0.1 $\%c$, and it is located at the leading and trailing edge. In the span-wise direction, elements are distributed uniformly and the number of elements is set to 100 over the semi-span. A prism layer is used to capture the viscous boundary layer. This layer consists of a number of structured elements that grow in size normal to the wing surface into the domain volume. The inflation layer has an initial height of $5 \times 10^{-6}c$ where it is grown 20 layers into the volume using an exponential growth law with a ratio of 1.2. The initial layer height is chosen so that $y^+ < 1$ at all nodes on the wing.

The numerical fluid flow simulations are performed using the computer code ANSYS FLUENT [34]. The implicit density-based solver is applied using the Roe flux-difference splitting (FDS) flux type. The spatial discretization schemes are set to second order for all variables, and the gradient information is found using the Green–Gauss node-based method. The residuals, which are the sum of the L^2 norms of all governing equations in each cell, are monitored and checked for convergence. For the high-fidelity model, a solution is considered to be converged if the residuals have dropped by six orders of magnitude, or the total number of iterations has reached 1,000. Also, the lift and drag coefficients are monitored for convergence.

A grid convergence study is conducted using the ONERA M6 wing [35]. The flow past the ONERA M6 wing is simulated at various grid resolutions at $Re_{\infty,c_{\text{mac}}} = 11.72 \times 10^6$, $M_\infty = 0.8395$, and angle of attack $\alpha = 3.06°$, where c_{mac} is the mean aerodynamic chord length. The flow conditions are selected to match the

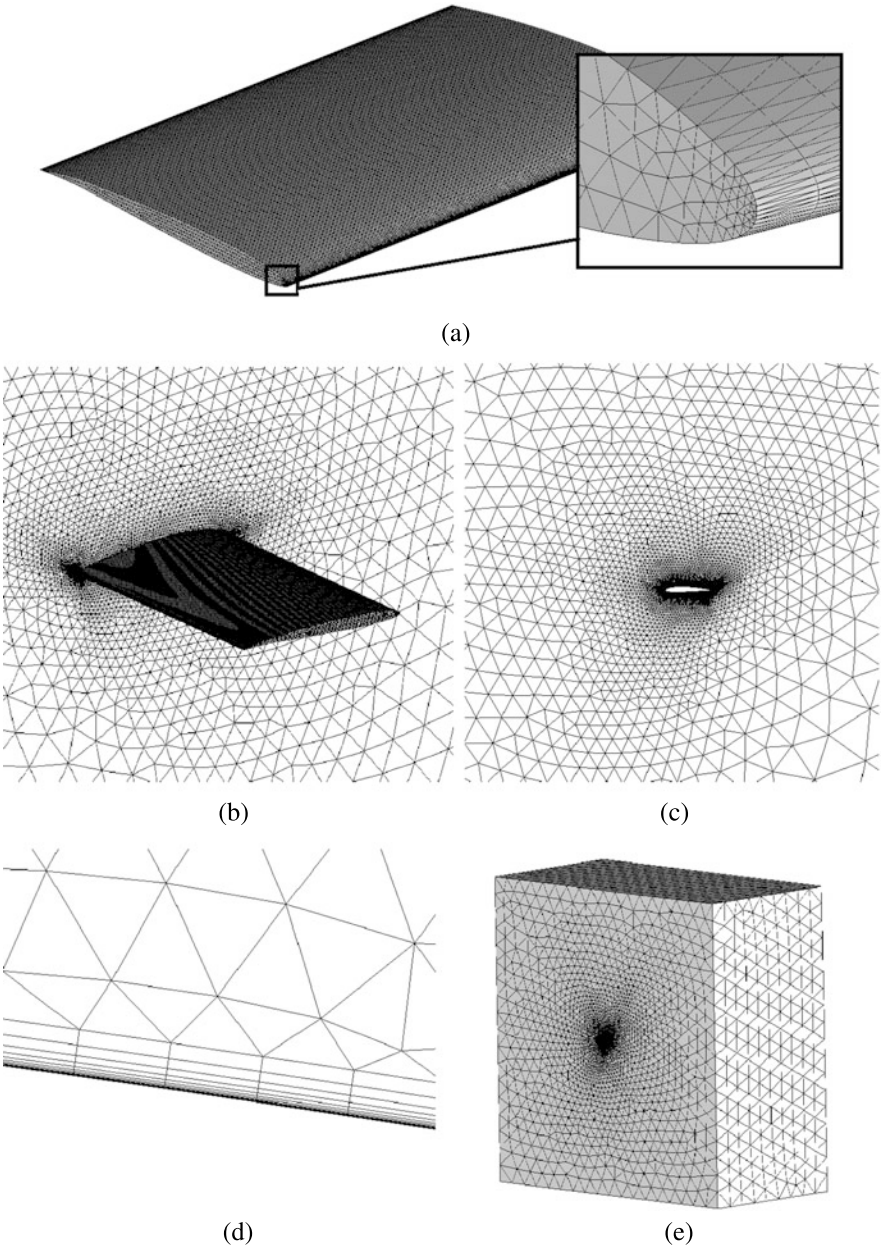


Fig. 12 Shell grid shown for all surfaces: (**a**) wing shell grid, (**b**)–(**c**) symmetry plane where the wing is placed, (**d**) prism layer applied close to the wing surface to capture the viscous boundary layer, (**e**) far-field volume

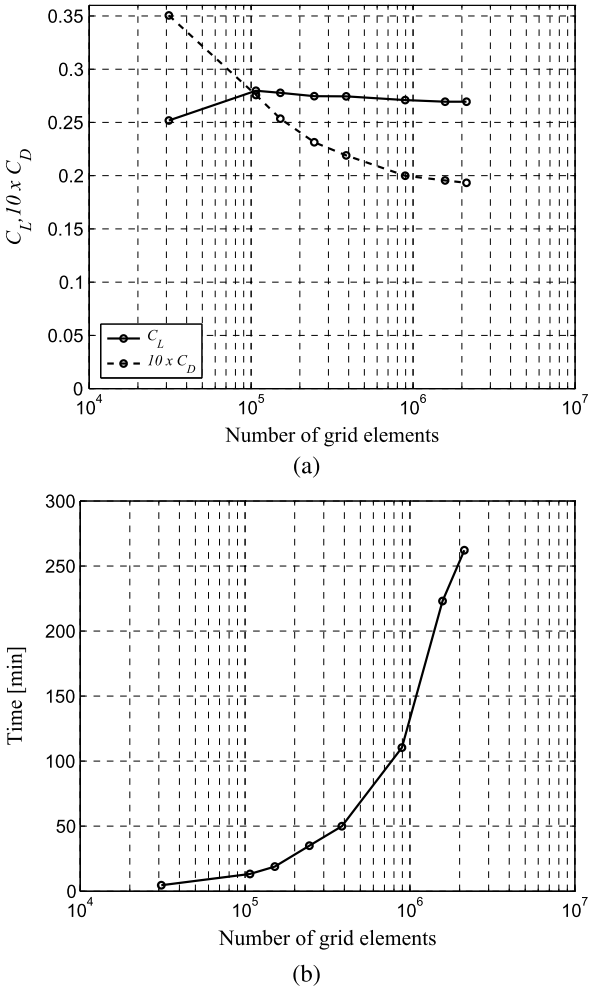


Fig. 13 Grid convergence study using the ONERA M6 wing at $M_\infty = 0.8395$ and angle of attack $\alpha = 3.06°$: (**a**) lift (C_L) and drag (C_D) coefficients versus number of grid elements, (**b**) simulation time versus number of grid elements

experimental flow conditions of an ONERA M6 wing experiment 2308 conducted by Schmitt and Charpin [36].

The grid convergence study, shown in Fig. 13a, revealed that 1,576,413 cells are needed for convergence in lift. The drag, however, can still be improved, as is evident from Fig. 13a, where convergence has not been reached due to limitations in the computational resources. However, we proceed with this grid as the high-fidelity model grid. The overall simulation time needed for one high-fidelity CFD simulation was around 223 min, as shown in Fig. 13b, executed on four Intel-i7-

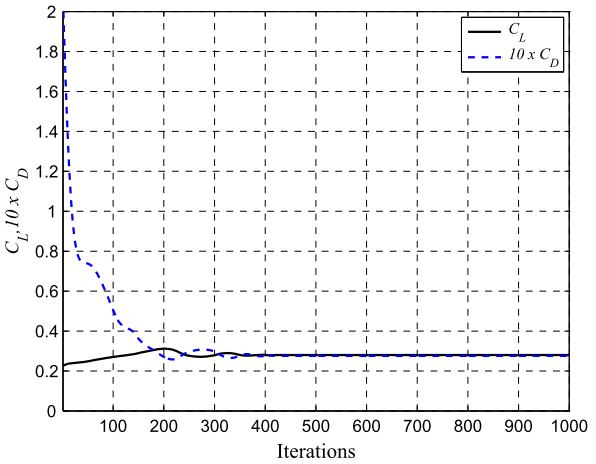


Fig. 14 Lift and drag coefficient convergence plot for low-fidelity model obtained in grid convergence study simulation using ONERA M6 wing at Mach number $M_\infty = 0.8395$ and angle of attack $\alpha = 3.06°$

2600 processors in parallel. This execution time is based on 1,000 solver iterations, where the solver terminated due to the maximum number of iterations limit.

5.2 Low-Fidelity Model

The low-fidelity model $c(\mathbf{x})$ is constructed in the same way as the high-fidelity model $f(\mathbf{x})$, but with a coarser grid discretization and with a relaxed convergence criterion. Referring back to the grid study of the previous section, and inspecting Fig. 13a, we make our selection for the coarse low-fidelity model. Based on time and accuracy with respect to lift and drag, we select the grid parameters that represent the second point from the left with 107,054 elements. The time taken to evaluate the low-fidelity model is 13.2 min on four Intel-i7-2600 processors in parallel.

Inspecting further the lift and drag convergence plot for the low-fidelity model in Fig. 14, we note that the solution has converged after about 500 iterations. The maximum number of iterations for the low-fidelity model is therefore set to 500 iterations. This reduces the overall simulation time to 6.6 min. The ratio of the simulation times of the high- and low-fidelity models in this case is high/low $= 223/6.6 \approxeq 34$. This is based on the solver using all 500 iterations in the low-fidelity model to obtain a solution.

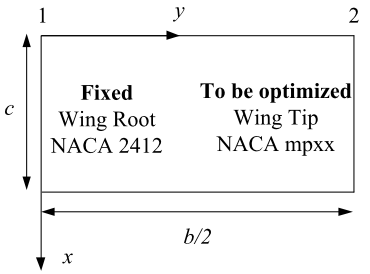


Fig. 15 A planform view of a constant chord rectangular wing constructed by two NACA airfoils. Each airfoil has its own set of design parameters

5.3 Surrogate Model

The low-fidelity CFD model c turns out to be very noisy. In order to alleviate the problem, a second-order polynomial approximation model [7] is constructed using $n_c = 50$ training points sampled using Latin hypercube sampling (LHS) [6] using the low-fidelity CFD model. The polynomial approximation model is defined as

$$\overline{c}(\mathbf{x}) = c_0 + \mathbf{c}_1^T \mathbf{x} + \mathbf{x}^T \mathbf{c}_2 \mathbf{x}, \tag{36}$$

where $\mathbf{c}_1 = [c_{1.1}\ c_{1.2}\ c_{1.3}]^T$ and $\mathbf{c}_2 = [c_{2.ij}]_{i,j=1,2,3}$. The coefficients c_0, $\mathbf{c}_1$, $\mathbf{c}_2$ are found by solving the linear regression problem

$$\overline{c}\left(\mathbf{x}^k\right) = c\left(\mathbf{x}^k\right), \tag{37}$$

where $k = 1, \ldots, n_c$. The resulting second-order polynomial model $\overline{c}$ has nice analytical properties, such as smoothness and convexity. The surrogate model is then constructed as described in Sect. 3.

5.4 Case Description

For demonstration purposes, an unswept and untwisted wing is considered. The wing is constructed by two NACA four-digit airfoils, located at the root and tip, as shown in Fig. 15. The root is fixed to the NACA 2412 airfoil and the tip airfoil is to be designed. The initial design $\mathbf{x}^{(0)}$ for the wing tip is chosen at random at the start of each optimization run. The normalized semi-wingspan is set as twice the wing chord length c as $(b/2) = 2c$. All other wing parameters are kept fixed. The design vector can be written as $\mathbf{x} = [m\ p\ t/c]^T$, where the variables represent the wing tip NACA four-digit airfoil parameters (see the Appendix).

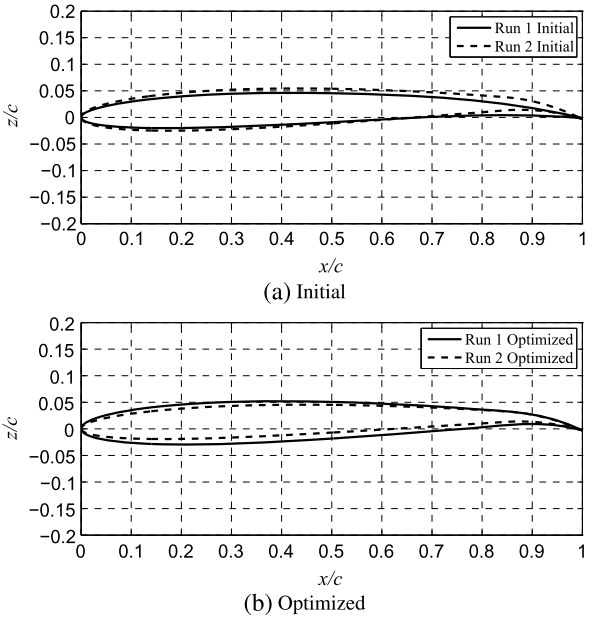
The objective is to maximize the lift coefficient $C_{L.f}$, subject to constraints on the drag coefficient $C_{D.f} \leq C_{D.\max} = 0.03$ and the wing tip normalized cross-sectional area $A \geq A_{\min} = 0.01$. The side constraints on the design variables are $0.02 \leq m \leq 0.03$, $0.7 \leq p \leq 0.9$ and $0.06 \leq t/c \leq 0.08$.

Table 3 Numerical comparison of Run 1 and Run 2, initial and optimized designs. The ratio of the high-fidelity model evaluation time to the low-fidelity time is 34

Variable	Initial		Optimized	
	Run 1	Run 2	Run 1	Run 2
m	0.0200	0.0259	0.0200	0.0232
p	0.7000	0.8531	0.8725	0.8550
t/c	0.0628	0.0750	0.0793	0.0600
C_L	0.2759	0.3426	0.3047	0.3388
C_D	0.0241	0.0344	0.0311	0.0307
C_L/C_D	11.4481	9.9593	9.7974	11.0358
A	0.0422	0.0505	0.0534	0.0404
N_c	–	–	50	50
N_f	–	–	8	7
Total cost	–	–	<10	<9

5.5 Results

Two optimization runs were performed, denoted as Run 1 and Run 2. The numerical results are given in Table 3, and the initial and optimized airfoil cross sections are shown in Fig. 16a and Fig. 16b, respectively.



(a) Initial

(b) Optimized

Fig. 16 A comparison of Run 1 and Run 2: (**a**) initial and (**b**) optimized designs. Run 1 is shown with a *solid lines* (–), and Run 2 with *dashed lines* (- -)

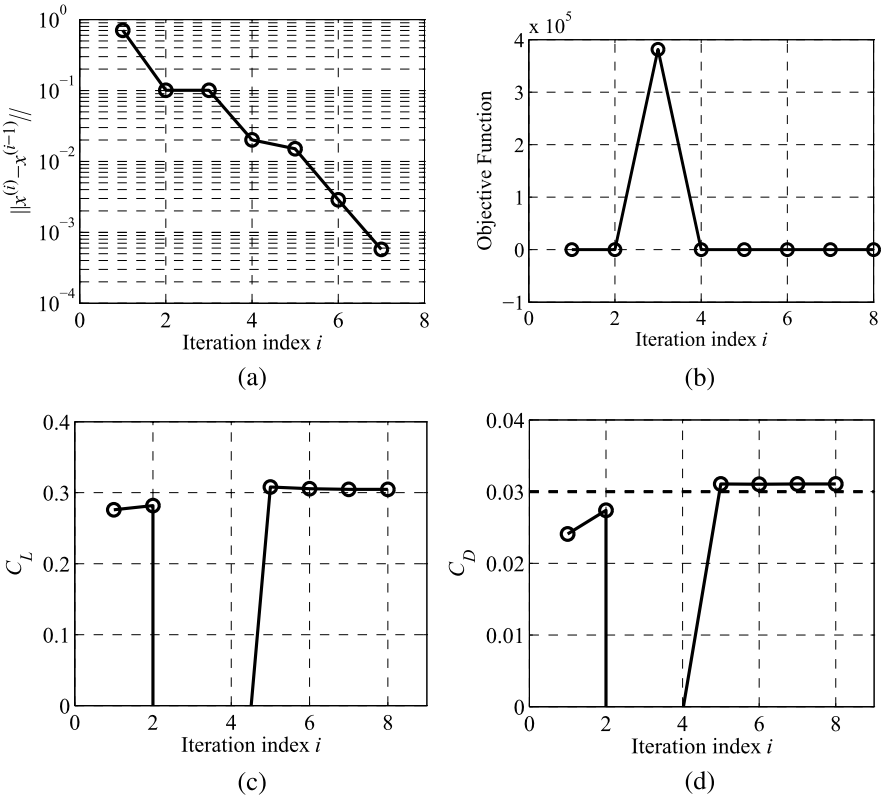
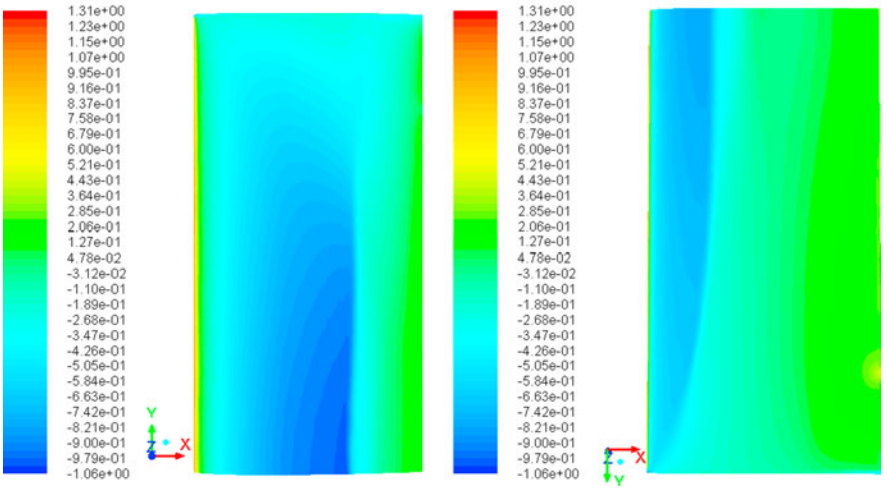


Fig. 17 Optimization history for Run 1: (**a**) convergence history, (**b**) evolution of the objective function, (**c**) evolution of lift coefficient, and (**d**) evolution of drag coefficient where *dashed line* (- -) is the drag constraint

In Run 1, the lift is increased by $+10$ % and the drag is pushed above its constraint at $C_{D.\max} = 0.03$, where the optimized drag coefficient is $C_D = 0.0311$. The drag constraint is violated slightly, or by $+4$ %, which is within the 5 % constraint tolerance band. The lift-to-drag ratio is decreased by -14 %. The SM algorithm requires less than 10 high-fidelity model evaluations, where 50 low-fidelity model evaluations (N_c) are used to create the approximation model and 8 high-fidelity model evaluations (N_f) are used for each design iteration. It is evident that the optimized wing tip airfoil is thicker as the normalized cross-sectional area is increased by $+26$ %, and the increased drag can be related to the increment in area. No change occurs in the camber m, but the location of the maximum camber p has moved slightly aft. The convergence history is shown in Fig. 17.

The initial design for Run 2 violates the drag constraint. The SM algorithm is, however, able to push the drag to its constraint limit, where the optimized drag coefficient is slightly violated, by $+2$ %. While the drag is decreased by -11 %, the lift is maintained and only drops by -1 %. As a result, the lift-to-drag ratio is increased



(a) Upper surface (b) Lower surface

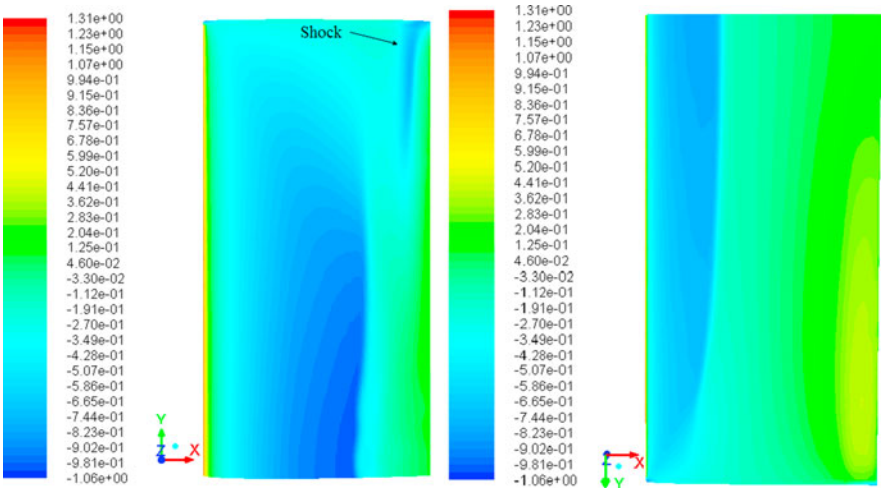
Fig. 18 Run 1 planform pressure coefficient contour plots of the initial design geometry. (**a**) The *upper surface* shows shocks at midsection of the wing; (**b**) the *lower surface* shows one shock at the leading edge

by $+11$ %. The SM algorithm requires less than 9 high-fidelity model evaluations (50 low-fidelity model evaluations used to create the approximation model and 7 high-fidelity model evaluations). The optimized wing tip airfoil is thinner than the initial design (the normalized cross-sectional area is reduced by -20 %). Small changes are made to the camber m and the maximum camber location p.

Comparing Runs 1 and 2, we note that, although they start from different initial designs, the optimized designs show similarities in two of three design variables, namely, the maximum camber m and maximum camber location p. The third, the airfoil thickness t/c, differs by approximately 2 %. The shock on the mid-wing has been moved aft, on both the upper and the lower surfaces (see Figs. 18, 19, 20, 21). Also, note that a second shock was formed near the tip on the upper surface. This causes the drag and the lift to increase.

6 Conclusions

A robust and computationally efficient optimization methodology for the design of aerodynamic surfaces is presented. The approach exploits a low-fidelity model that is corrected using the space mapping technique to create a fast and reliable prediction tool (the surrogate) that is subsequently used to yield an approximate optimum design of the expensive, high-fidelity model at low CPU cost. A space mapping correction is applied both to the objectives and constraints in a two-stage process with a quasi-global space mapping alignment supplemented by a local one that ensures



(a) Upper surface (b) Lower surface

Fig. 19 Run 1 planform pressure coefficient contour plots of the optimized design geometry. (**a**) The *upper surface* shows two shocks, one at midsection of the wing and one close to the wing tip at the trailing edge. (**b**) The *lower surface* shows one shock at the leading edge

perfect alignment between the surrogate and the high-fidelity model. Applications for transonic airfoil and wing design are demonstrated with the optimized designs obtained at a computational cost corresponding to a few high-fidelity model evaluations.

Acknowledgements This work was funded in part by The Icelandic Research Fund for Graduate Students, grant ID: 110395-0061.

Appendix

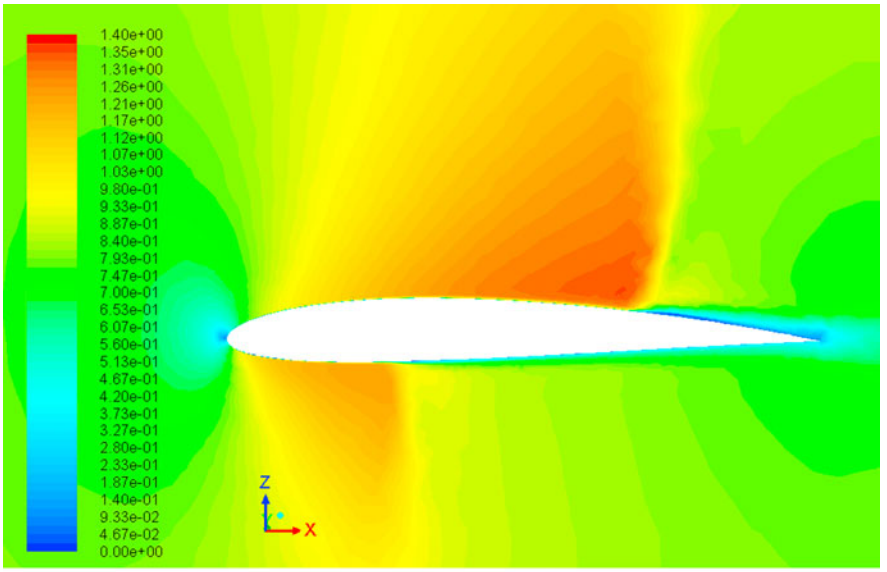
The NACA four-digit airfoils are denoted by convention as NACA $mpxx$, where m is the maximum ordinate of the mean camber line as a percentage of the chord, p is the chordwise position in tens of percentages of the maximum ordinate, and xx is the thickness-to-chord ratio in percentages of the chord (t/c). The NACA airfoils are constructed by combining a thickness function $z_t(x)$ with a mean camber line function $z_c(x)$ [25]. The x and z coordinates are

$$x_{u,l} = x \mp z_t \sin\theta, \tag{38}$$

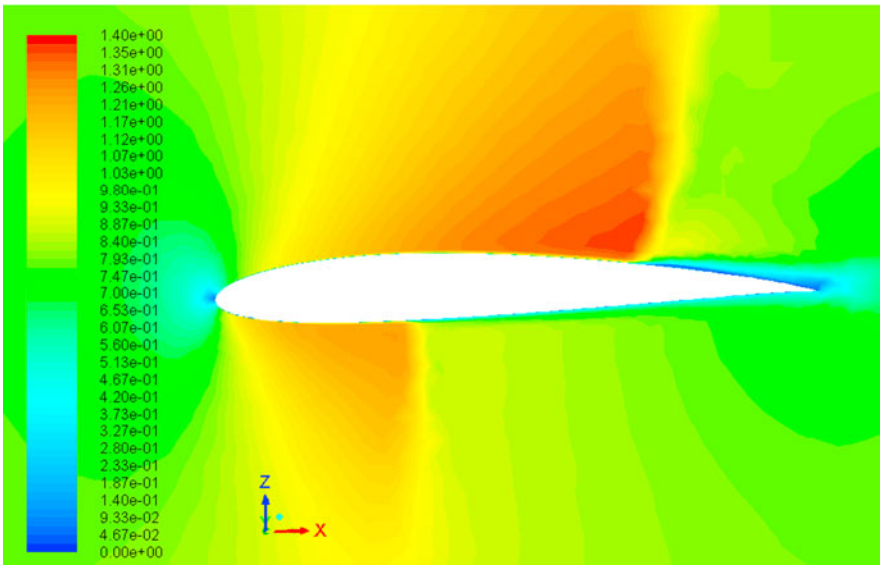
$$z_{u,l} = z_c \pm z_t \cos\theta, \tag{39}$$

where u and l are the upper and lower surfaces, respectively, and

$$\theta = \tan^{-1}\left(\frac{dz_c}{dx}\right), \tag{40}$$

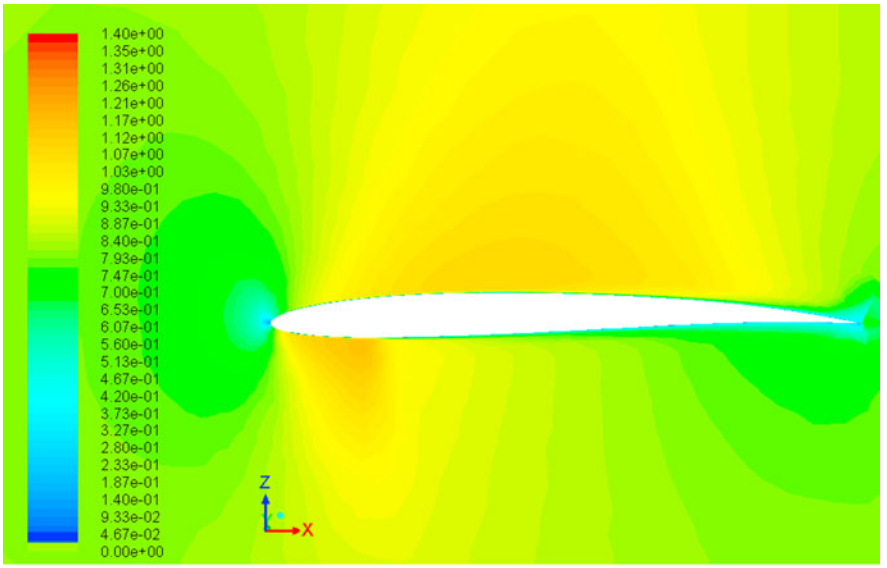


(a) Initial

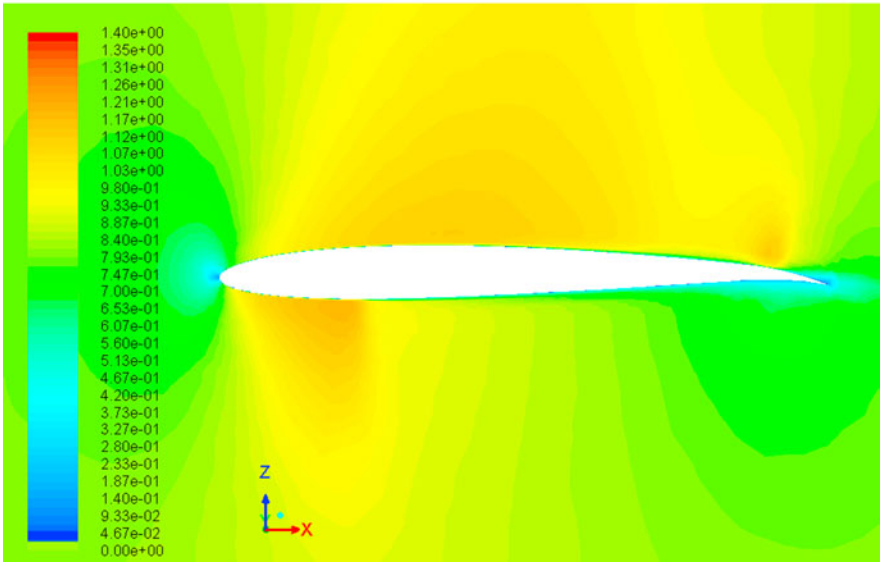


(b) Optimized

Fig. 20 Run 1 Mach number contour plot at $y/(b/2) = 0.2$, where $M_\infty = 0.8395$ and angle of attack $\alpha = 0°$. (**a**) Initial design, (**b**) optimized design



(a) Initial



(b) Optimized

Fig. 21 Run 1 Mach number contour plot at $y/(b/2) = 0.8$, where $M_\infty = 0.8395$ and angle of attack $\alpha = 0°$: (**a**) initial design, (**b**) optimized design. Notice the shock at the trailing edge

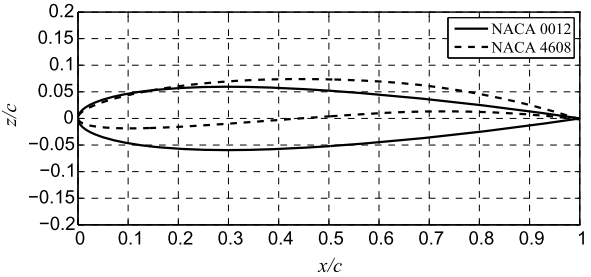


Fig. 22 Examples of two airfoil sections generated with the NACA four-digit method. NACA 0012 ($m = 0, p = 0, t/c = 0.12$) is shown by *solid line* (–); NACA 4608 ($m = 0.04$, $p = 0.6, t/c = 0.08$) is shown by a *dashed line* (- -)

is the mean camber line slope. The NACA four-digit thickness distribution is given by

$$z_t = t\left(a_0 x^{1/2} - a_1 x a_2 x^2 + a_3 x^3 - a_4 x^4\right), \tag{41}$$

where $a_0 = 1.4845, a_1 = 0.6300, a_2 = 1.7580, a_3 = 1.4215, a_4 = 0.5075$, and t is the maximum thickness. The mean camber line is given by

$$z_c = \begin{cases} \frac{m}{p^2}(2px - x^2), & x < p, \\ \frac{m}{(1-p)^2}(1 - 2p + 2px - x^2), & x \geq p. \end{cases} \tag{42}$$

Examples of airfoils generated with the NACA four-digit method are shown in Fig. 22.

References

1. Jameson, A.: Aerodynamic design via control theory. J. Sci. Comput. **3**, 233–260 (1988)
2. Braembussche, R.A.: Numerical Optimization for Advanced Turbomachinery Design (2008)
3. Dumas, L.: CFD-Based Optimization for Automotive Aerodynamics (2008)
4. Percival, S., Hendrix, D., Noblesse, F.: Hydrodynamic optimization of ship hull forms. Appl. Ocean Res. **23**(6), 337–355 (2001)
5. Queipo, N.V., Haftka, R.T., Shyy, W., Goel, T., Vaidyanathan, R., Tucker, P.K.: Surrogate-based analysis and optimization. Prog. Aerosp. Sci. **41**(1), 1–28 (2005)
6. Forrester, A.I.J., Keane, A.J.: Recent advances in surrogate-based optimization. Prog. Aerosp. Sci. **45**(1–3), 50–79 (2009)
7. Koziel, S., Ciaurri, D.E., Leifsson, L.: Surrogate-based methods. In: Computational Optimization and Applications in Engineering and Industry, vol. 359. Springer, Berlin (2011)
8. Wild, S.M., Regis, R.G., Shoemaker, C.A.: Orbit: optimization by radial basis function interpolation in trust-regions. SIAM J. Sci. Comput. **30**(6), 3197–3219 (2008)
9. Simpson, T.W., Poplinski, J.D., Koch, P.N., Allen, J.K.: Metamodels for computer-based engineering design: survey and recommendations. Eng. Comput. **17**(2), 129–150 (2001)
10. Journel, A.G., Huijbregts, C.J.: Mining Geostatistics. Academic Press, London (1978)
11. O'Hagan, A., Kingman, J.F.C.: Curve fitting and optimal design for prediction. J. R. Stat. Soc. Ser. B Methodol., 1–42 (1978)

12. Haikin, S.: Neural Networks: A Comprehensive Foundation. Prentice-Hall, Englewood Cliffs (1998)
13. Minsky, M., Papert, S.: Perceptrons: An Introduction to Computational Geometry. MIT Press, Cambridge (1969)
14. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Stat. Comput. **14**(3), 199–222 (2004)
15. Søndergaard, J.: Optimization using surrogate models by the space mapping technique. Ph.D. thesis, Technical University of Denmark, Informatics and mathematical modelling (2003)
16. Leifsson, L., Koziel, S.: Variable-fidelity aerodynamic shape optimization. In: Computational Optimization, Methods and Algorithms, vol. 356. Springer, Berlin (2011)
17. Bandler, J.W., Cheng, Q.S., Dakroury, S.A., Mohamed, A.S., Bakr, M.H., Madsen, K., Sondergaard, J.: Space mapping: the state of the art. IEEE Trans. Microw. Theory Tech. **52**(1), 337–361 (2004)
18. Alexandrov, N.M., Lewis, R.M.: An overview of first-order model management for engineering optimization. Optim. Eng. **2**(4), 413–430 (2001)
19. Koziel, S., Cheng, Q., Bandler, J.: Space mapping. IEEE Microw. Mag. **9**(6), 105–122 (2008)
20. Echeverria, D., Hemker, P.W.: Space mapping and defect correction. Comput. Methods Appl. Math. **5**(2), 107–136 (2005)
21. Koziel, S.: Shape-preserving response prediction for microwave design optimization. IEEE Trans. Microw. Theory Tech. **58**(11), 2829–2837 (2010)
22. Robinson, T.D., Willcox, K.E., Eldred, M.S., Haimes, R.: Multifidelity optimization for variable complexity design. In: Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Portsmouth, VA (2006)
23. Koziel, S., Leifsson, L.: Knowledge-based airfoil shape optimization using space mapping. In: 30th AIAA Applied Aerodynamics Conference, New Orleans, Louisiana, June 25–28 (2012), AIAA 2012-3016
24. Jonsson, E., Koziel, S., Leifsson, L.: Transonic wing optimization by variable-resolution modeling and space mapping. In: 2nd Int. Conf. on Simulation and Modeling Methodologies, Technologies, and Applications (SIMULTECH), Rome, Italy, July 28–31 (2012)
25. Abbott, I.H., Von Doenhoff, A.E.: Theory of Wing Sections: Including a Summary of Airfoil Data. Dover Publications, New York (1959)
26. Hicks, R.M., Henne, P.A.: Wing design by numerical optimization. J. Aircr. **15**(7), 407–412 (1977)
27. Derksen, R.W., Rogalsky, T.: Bezier-PARSEC: an optimized aerofoil parameterization for design. Adv. Eng. Softw. **41**, 923–930 (2010)
28. Leoviriyakit, K., Kim, S., Jameson, A.: Viscous aerodynamic shape optimization of wings including planform variables. In: 21st AIAA Applied Aerodynamics Conference, Orlando FL (2003), AIAA Paper AIAA-2003-3791
29. Koziel, S., Cheng, Q.S., Bandler, J.W.: Implicit space mapping with adaptive selection of preassigned parameters. IET Microw. Antennas Propag. **4**(3), 361–373 (2010)
30. Koziel, S., Bandler, J.W., Madsen, K.: A space mapping framework for engineering optimization: theory and implementation. IEEE Trans. Microw. Theory Tech. **54**(10), 3721–3730 (2006)
31. Zhu, J., Bandler, J.W., Nikolova, N.K., Koziel, S.: Antenna optimization through space mapping. IEEE Trans. Antennas Propag. **55**(3), 651–658 (2007)
32. Tannehill, J.C., Anderson, D.A., Pletcher, R.H.: Computational Fluid Mechanics and Heat Transfer. Taylor & Francis Group, London (1997)
33. ANSYS: ANSYS ICEM CFD. Southpointe 275 Technology Drive, Canonburg, PA 15317, release 13.0 edition, November (2010)
34. ANSYS: ANSYS FLUENT Theory Guide. ANSYS, Southpointe 275 Technology Drive, Canonburg PA 15317, release 13.0 edition, November (2010).

35. NASA: ONERA M6 wing validation case (2008). http://www.grc.nasa.gov/WWW/wind/valid/m6wing/m6wing.html
36. Schmitt, V., Charpin, F.: Pressure distributions on the ONERA-M6-wing at transonic Mach numbers. Experimental Data Base for Computer Program Assessment. Report of the Fluid Dynamics Panel Working Group 04, AGARD AR 138, May (1979)

Efficient Robust Design with Stochastic Expansions

Yi Zhang and Serhat Hosder

Abstract This chapter describes the application of a computationally efficient uncertainty quantification approach, non-intrusive polynomial chaos (NIPC)-based stochastic expansions, for robust design under mixed (aleatory and epistemic) uncertainties and demonstrates this technique on robust design of a beam and on robust aerodynamic optimization. The approach utilizes stochastic response surfaces obtained with NIPC methods to approximate the objective function and the constraints in the optimization formulation. The objective function includes the stochastic measures, which are minimized simultaneously to ensure the robustness of the final design to both aleatory and epistemic uncertainties. The results of the optimization case studies show the computational efficiency and accuracy of the robust design with stochastic expansions, which may be applied to any stochastic optimization problem in science and engineering.

Keywords Aerodynamics · Optimization · Uncertainty quantification · Robust design · Stochastic expansions · Computational fluid dynamics

1 Introduction

Uncertainties are generally ubiquitous in the analysis and design of highly complex engineering systems, such as aerospace systems. Uncertainties can arise due to ignorance, lack of knowledge, and incomplete information in physical modeling (e.g., epistemic uncertainty in turbulence models) and from inherent variations in the systems (e.g., aleatory uncertainty in operating conditions). It is important to consider these uncertainties in engineering design. Robust design [1, 2] is a methodology for improving the quality of a product by minimizing the impact of uncertainties on the product performance. The objective is to optimize the mean performance while minimizing the variation of performance caused by various uncertainties. Many studies of robust design have been investigated in the past [3–12]. A comprehensive survey of robust optimization approaches is given by Beyer and Sendhoff [13].

Y. Zhang · S. Hosder (✉)
Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, Rolla, MO, USA
e-mail: hosders@mst.edu

S. Koziel, L. Leifsson (eds.), *Surrogate-Based Modeling and Optimization*,
DOI 10.1007/978-1-4614-7551-4_11,

One very important component of robust design is the uncertainty quantification (UQ), which may increase the computational expense of the design process significantly compared to the computational work of deterministic optimization, especially when high-fidelity analysis tools are used to improve accuracy. Therefore, it is important to develop and implement computationally efficient robust design methodologies while keeping the desired accuracy level in the optimization process.

The goal of UQ is to determine how random variation (aleatory) and lack of knowledge (epistemic) affect the sensitivity, performance, or reliability of the system that is being modeled. Various studies have been made on the topic of propagating aleatory uncertainty through Monte Carlo sampling (MCS) [14–18], expansion-based methods (e.g., Taylor series [19–22] and perturbation methods [23–25]) and non-intrusive polynomial chaos (NIPC) expansions [26–28], and propagating epistemic uncertainty through interval analysis and evidence theory [29–34].

MCS is the most comprehensive (but expensive) UQ approach for evaluating statistical moments and the reliability and quality of the system response. It is a method for iteratively evaluating a deterministic model using sets of random numbers as inputs. This method is often used when the model is complex or nonlinear, or when it involves a large number of uncertain parameters. M.G. Cox et al. [16] worked on using MCS to determine the probability density function (PDF) of the output quantities. L.Y. Zhao et al. investigated UQ of a flapping airfoil with stochastic velocity deviations by using a classic Monte Carlo method to numerically investigate the responses of the time-averaged thrust coefficient and the propulsive efficiency with respect to a stochastic flight velocity deviation under Gaussian distributions [17, 18]. Y.P. Ju et al. conducted studies on multi-point robust design optimization of wind turbine airfoils under geometric uncertainty where the MCS technique was used for simulating the geometric uncertainty in the robust optimization [35]. Although MCS is the most popular sampling-based method, it requires thousands of computational simulations (e.g., computational fluid dynamics (CFD) and finite element analysis (FEA)) to obtain accurate results. It is extremely expensive and cannot be made affordable for complex engineering simulations, so it is often used as a benchmark for verification of UQ analysis when other methods are used.

Expansion-based UQ is used to estimate the statistical moments (e.g., mean, variance, etc.) of the system response with a small perturbation to simulate the effect of the input uncertainty. The Taylor series and perturbation method are two main expansion-based UQ approaches. The Taylor series is a series expansion of a function about a point that is used to approximate a function with a Taylor polynomial. For example, the first-order reliability method (FORM) uses the first-order Taylor expansion (linearization) to approximate the uncertainty in the output [19]. There have also been some studies on Taylor series expansion techniques and applications in physics [20–22]. The perturbation method is used to find an approximate solution to a problem which cannot be solved by traditional analytical methods. It allows the simplification of complex mathematical problems [23–25]. Both Taylor series and perturbation methods have advantages when dealing with relatively small input variability and outputs that do not express high nonlinearity. However, most real-life problems require much more difficult mathematical models, such as nonlinear

differential equations. Therefore, Taylor series and perturbation methods will not be a good option for uncertainty propagation in these cases.

The NIPC expansion is a spectral-based technique for UQ that has been used recently for numerous physical models, such as elasticity and fluid mechanics [36–38]. Some studies conducted by Eldred et al. [27, 28] introduced efficient stochastic expansions based on NIPC for UQ. In their study, they used Legendre and Hermite orthogonal polynomials to model the effect of uncertain variables described by uniform and normal probability distributions, respectively, and used Legendre orthogonal polynomials to model the effect of epistemic uncertainties. The accuracy and the computational efficiency of the NIPC method applied to stochastic problems with multiple uncertain input variables were investigated by Hosder et al. [39–41].

The non-probabilistic approaches which are used for epistemic UQ include interval analysis and evidence theory. Several studies have been conducted on epistemic uncertainty propagation [29–31]. The simplest way to propagate epistemic uncertainty is by interval analysis [32–34]. In interval analysis, it is assumed that nothing is known about the uncertain input variables except that they lie within certain intervals. L.P. Swiler and T.L. Paez [32, 33] examined three methods in propagating epistemic uncertainty: interval analysis, Dempster–Shafer evidence theory, and second-order probability, and demonstrated examples of their use on a problem in structural dynamics. They also examined the use of surrogate methods in epistemic analysis, both surrogate-based optimization in interval analysis and the use of polynomial chaos expansions to provide upper and lower bound approximations. From their studies, it was proved that interval analysis can be effective in the quantification of epistemic uncertainty.

Recently, there have been some studies investigating the topic of mixed (aleatory and epistemic) uncertainty propagation. Eldred et al. [42] proposed using second-order probability for estimating the effect of mixed uncertainties. This method was used to separate the aleatory and epistemic uncertainties into inner and outer sampling loops, respectively. Moreover, they applied this method to a cantilever beam design problem which was represented by two simple analytical functions. They utilized these analytical functions to represent ideal test cases, since they were inexpensive to evaluate. Therefore, this study provided an analytical reference for validating codes used for mixed aleatory and epistemic UQ. Bettis and Hosder applied the NIPC approach to the propagation of mixed uncertainties in hypersonic reentry problems [41, 43].

Most of the previous stochastic design studies focused on optimization under aleatory uncertainties which utilized different approaches for uncertainty propagation [3–12, 44]. Among these studies, Eldred [44] formulated and investigated design under aleatory uncertainty with stochastic expansions. Dodson and Parks utilized polynomial chaos expansions for robust airfoil design under aleatory input uncertainties [45]. B.D. Youn et al. also developed a robust design optimization with epistemic uncertainty. They proposed a new metric for product quality loss which was defined for epistemic uncertainty using the analogy between the probability and possibility theories. For the epistemic uncertainty, the maximum likely value and equivalent variation were employed to define the new metric for the product

quality loss in three different types of robust objectives [46]. A number of robust design studies have considered both aleatory and epistemic uncertainties, such as the work by Eldred [47], and Du et al. [48], who used a double-loop Monte Carlo sampling approach to determine the statistics of the response given in their model problems.

In this chapter, we describe a computationally efficient approach for robust aerodynamic optimization under aleatory (inherent) and epistemic (model-form) uncertainties using stochastic expansions based on the NIPC method [49]. In the context of aerodynamic shape optimization, robust design implies that the performance (such as the lift-to-drag ratio) of the final configuration should be insensitive to the uncertainties in the operating conditions (e.g., free-stream Mach number). Furthermore, the final design should be relatively insensitive to the physical modeling uncertainties in the computational tools used for aerodynamic analysis such as the computational fluid dynamics (CFD) codes.

The chapter is organized as follows. Section 2 describes the details of the robust design formulation with stochastic expansions. In Sect. 3, the methodology is applied to a model problem involving the design of a beam, and Sect. 4 describes an application to airfoil design at transonic flow conditions. The chapter is summarized in Sect. 5.

2 Robust Design with Stochastic Expansions

This section gives the details of different robustness measures for a system depending on the input uncertainty type, as well as the utilization of stochastic expansions in robust optimization.

2.1 Formulation of Robust Optimization

Following Du et al. [48], we describe the following robust measures: (1) purely aleatory (inherent) input uncertainty, (2) purely epistemic input uncertainty, and (3) mixed (aleatory and epistemic) input uncertainty.

2.1.1 Aleatory Uncertainties Only

If there are only aleatory uncertainties as input variables, the response R can be described as a function of $\mathbf{S}_a = (S_{a_1}, S_{a_2}, \ldots, S_{a_{N_a}})$, which is the vector consisting of N_a aleatory uncertainties; this vector can include both aleatory design variables ($\mathbf{X}_a$) and aleatory parameters ($\mathbf{P}_a$). In a design study, the aleatory uncertainty can be imposed on the design variables through the statistical distribution parameters that define them (e.g., mean and variance), which vary in the design space. Figure 1

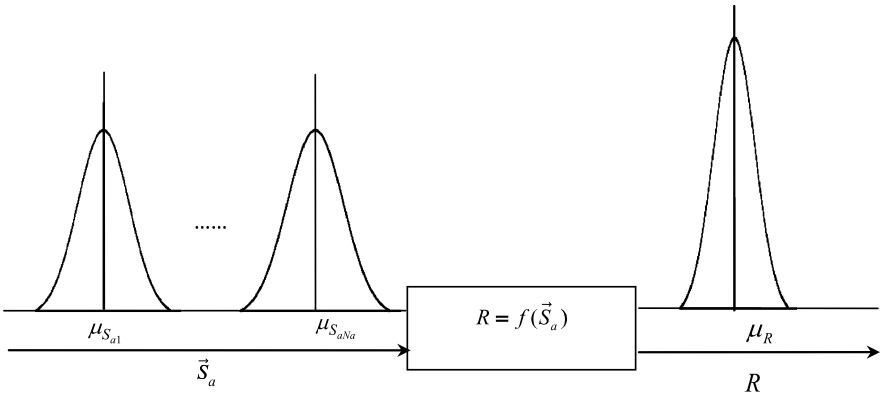


Fig. 1 Robustness estimation of response in the presence of aleatory uncertainties only

shows the propagation of input aleatory uncertainties through the simulation code and the uncertainty of the response, $R = f(\mathbf{S}_a)$. For a probabilistic output uncertainty, the mean and the variance of R can be calculated by

$$\mu_R = E(R) = \int_{\Omega} R(\mathbf{S}_a)\rho(\mathbf{S}_a)\, d\mathbf{S}_a, \tag{1}$$

$$\sigma_R^2 = E\left[(R - \mu_R)^2\right] = \int_{\Omega} \left(R(\mathbf{S}_a) - \mu_R\right)^2 \rho(\mathbf{S}_a)\, d\mathbf{S}_a, \tag{2}$$

where $\rho(\mathbf{S}_a)$ represents the joint PDF of $\mathbf{S}_a$, and Ω stands for the support region of $\mathbf{S}_a$. For this case, the variance (or the standard deviation, σ) of R is considered as the robustness measure.

2.1.2 Epistemic Uncertainties Only

If there are only epistemic uncertainties as input, the response will be a function of epistemic uncertainty vector $\mathbf{S}_e = (S_{e_1}, S_{e_2}, \ldots, S_{e_{N_e}})$, which may include epistemic design variables ($\mathbf{X}_e$) and epistemic parameters ($\mathbf{P}_e$) in general. In a design study, the epistemic uncertainty can be imposed on the design variables through the parameters that define them (e.g., average and the limits of the interval), which vary in the design space. The relationship between input epistemic uncertainties and response $R = f(\mathbf{S}_e)$ is shown in Fig. 2. The midpoint ($\overline{R}$) and width (δR) of interval R are the most relevant statistics of response R for this case, and are given by

$$\overline{R} = \frac{1}{2}(R_L + R_U), \tag{3}$$

$$\delta R = R_U - R_L, \tag{4}$$

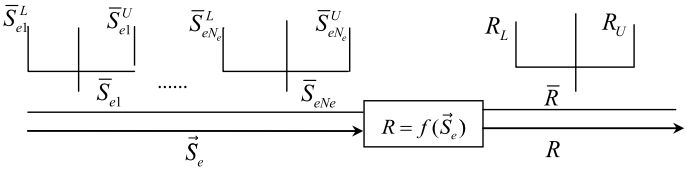


Fig. 2 Robustness estimation of response in the presence of epistemic uncertainties only

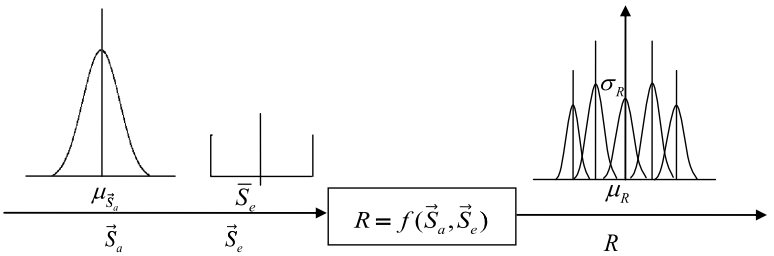


Fig. 3 Robustness estimation of response in the presence of mixed uncertainties

respectively, where R_U and R_L represent the upper bound and lower bound of R. For this case, the robustness of the response is assessed by δR. For robust optimization, δR should be as low as possible, while $\overline{R}$ is equal to the desired value.

2.1.3 Mixed Uncertainties (Both Aleatory and Epistemic Uncertainties)

When both aleatory uncertainties $\mathbf{S}_a = (S_{a_1}, \ldots, S_{a_{N_a}})$ and epistemic uncertainties $\mathbf{S}_e = (S_{e_1}, S_{e_2}, \ldots, S_{e_{N_e}})$ exist as input variables, the response R becomes a function of both types of uncertainty, $R = f(\mathbf{S}_a, \mathbf{S}_e)$, as shown in Fig. 3.

For this case, the uncertainty of R will be in the form of a family of probability distributions, each due to the aleatory input uncertainties at a fixed value of the epistemic input uncertainty vector. The intervals at each probability level will reflect the effect of epistemic uncertainties on R. The average mean value of R is calculated by

$$\overline{\mu}_R = \frac{1}{2}\left(\mu_R^{\max} + \mu_R^{\min}\right), \tag{5}$$

where $\mu_R^{\max}$ and $\mu_R^{\min}$ are the maximum and minimum means of response R, respectively. The average value of the standard deviation of R ($\overline{\sigma}_R$) is obtained by

$$\overline{\sigma}_R = \frac{1}{2}\left(\sigma_R^{\max} + \sigma_R^{\min}\right), \tag{6}$$

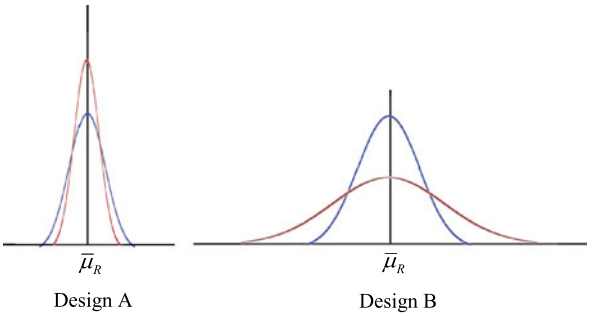


Fig. 4 Robustness assessment of mixed uncertainty design

where $\sigma_R^{\max}$ and $\sigma_R^{\min}$ are the maximum and minimum standard deviations of response R, respectively. The difference between $\sigma_R^{\max}$ and $\sigma_R^{\min}$ is computed by

$$\delta\sigma_R = \sigma_R^{\max} - \sigma_R^{\min}. \tag{7}$$

In a design study, the average standard deviation $\overline{\sigma}_R$ can be used as a robustness measure for aleatory input uncertainties ($\mathbf{S}_a$), whereas the standard deviation difference $\delta\sigma_R$ can be used as the robustness measure for epistemic uncertainties ($\mathbf{S}_e$). Note that one may also consider alternative measures for robustness to aleatory input uncertainties in the presence of mixed uncertainties. One approach will be to consider the maximum value of the standard deviation as a conservative measure, which in turn can be used in the robust optimization formulation described below.

2.1.4 Robust Optimization Formulation Under Mixed Uncertainties

To achieve a robust design in the presence of aleatory and epistemic uncertainties, both a lower value of $\overline{\sigma}_R$ and a lower value of $\delta\sigma_R$ are desired. To illustrate this, let us consider two designs (A and B) with performances (i.e., responses) having two different families of probability distributions represented by one blue and one red curve in Fig. 4. From these distributions, it is obvious that $\overline{\sigma}_R$ of design A is less than that of design B, which indicates that design A is more robust than design B when only randomness of the input is considered. Now comparing $\delta\sigma_R$ for the two designs, it can be seen that design A has a smaller difference between the distribution variances, indicating that it is also more robust to epistemic uncertainties. From the above discussion and following the formulation of Du et al. [48], a composite (weighted sum) objective function for robust optimization under mixed uncertainties can be written as

$$F(\mathbf{X}_d, \mathbf{S}_a, \mathbf{S}_e) = w_1\overline{\mu}_R + w_2\overline{\sigma}_R + w_3\delta\sigma_R, \tag{8}$$

where $\mathbf{X}_d$ is the deterministic design variable vector, $\mathbf{S}_a = \{\mathbf{X}_a, \mathbf{P}_a\}$, and $\mathbf{S}_e = \{\mathbf{X}_e, \mathbf{P}_e\}$. The values of the weight factors w_1, w_2, and w_3 should be chosen based on

the emphasis on the contribution of each term to the objective function by also considering the order of magnitude of each term. While minimizing F, a feasible design should also satisfy the inequality constraints $g_i(\mathbf{X}_d, \mathbf{S}_a, \mathbf{S}_e)$ $(i = 1, 2, \ldots, N_g)$ and the side constraints for aleatory design variables $\mathbf{X}_a$ (specified by the lower and the upper limits of the mean of the each aleatory variable), epistemic design variables $\mathbf{X}_e$ (specified by the lower and the upper limits of the epistemic variable), and the deterministic design variables $\mathbf{X}_d$. Under mixed uncertainties, a conservative form of the satisfaction of the inequality constraints can be written as $\mu_{gi}^{\max} + \beta\sigma_{gi}^{\max} \leq 0$, where $\mu_{g_i}^{\max}$ and $\sigma_{g_i}^{\max}$ are the maximum of the mean and the maximum of the standard deviation of the constraint function g_i, respectively. Here β_i is a positive constant which denotes the probability of constraint satisfaction.

In summary, the overall formulation for robust design optimization under aleatory and epistemic uncertainties can be written as:

$$
\begin{aligned}
&\text{Minimize} \quad && w_1\overline{\mu}_R + w_2\overline{\sigma}_R + w_3\delta\sigma_R \\
&\text{S.t.} \quad && \mu_{g_i}^{\max} + \beta_i\sigma_{g_i}^{\max} \leq 0, \quad i = 1, 2, \ldots, N_g \\
& && X_{d_j}^L \leq X_{d_j} \leq X_{d_j}^U, \quad j = 1, 2, \ldots, N_d \\
& && X_{e_k}^L \leq X_{e_k} \leq X_{e_k}^U, \quad k = 1, 2, \ldots, N_e \\
& && \mu_{X_{a_m}}^L \leq \mu_{X_{a_m}} \leq \mu_{X_{a_m}}^U, \quad m = 1, 2, \ldots, N_a.
\end{aligned}
\tag{9}
$$

2.2 Stochastic Expansions for Response Surface Modeling

For the robust optimization methodology described in this chapter, we adopt stochastic expansions obtained with non-intrusive polynomial chaos (NIPC) due to its computational efficiency and accuracy in aleatory and epistemic uncertainty propagation, as shown in the previous studies [43, 50]. The stochastic expansions are used as response surfaces (i.e., surrogates of the response) in the optimization procedure and are used to approximate the stochastic objective function or the constraint functions. In the robust optimization problems, we use two different NIPC approaches, point-collocation NIPC and quadrature-based NIPC. Below we give the description of these NIPC methods.

2.2.1 Non-intrusive Polynomial Chaos

Non-intrusive polynomial chaos is derived from polynomial chaos theory, which is based on the spectral representation of the uncertainty. An important aspect of spectral representation of uncertainty is that one may decompose a random function (or variable) into separable deterministic and stochastic components. For example, for any response variable (i.e., R) in a stochastic optimization problem, one can

write

$$R(\boldsymbol{\xi}) \approx \sum_{j=0}^{P} \alpha_j \Psi_j(\boldsymbol{\xi}), \tag{10}$$

where α_j is the coefficient of each term in the expansion, and $\Psi_j(\boldsymbol{\xi})$ is the random basis function corresponding to the jth mode and is a function of the n-dimensional random variable vector $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_n)$, which has a specific probability distribution. In theory, the polynomial chaos expansion given by Eq. (10) should include an infinite number of terms; however, in practice a discrete sum is taken over a number of output modes. For a total order expansion, the number of output modes is given by

$$N_t = P + 1 = \frac{(n+p)!}{n!p!}, \tag{11}$$

which is a function of the order of polynomial chaos (p) and the number of random dimensions (n). The basis function ideally takes the form of a multidimensional Hermite polynomial to span the n-dimensional random space when the input uncertainty is Gaussian (unbounded), which was first used by Wiener [51] in his original work of polynomial chaos. To extend the application of the polynomial chaos theory to the propagation of continuous non-normal input uncertainty distributions, Xiu and Karniadakis [52] used a set of polynomials known as the Askey scheme to obtain the "Wiener–Askey generalized polynomial chaos." The Legendre and Laguerre polynomials, which are among the polynomials included in the Askey scheme, are optimal basis functions for bounded (uniform) and semibounded (exponential) input uncertainty distributions, respectively, in terms of the convergence of the statistics. The multivariate basis functions can be obtained from the product of univariate orthogonal polynomials (see Eldred et al. [27]). If the probability distribution of each random variable is different, then the optimal multivariate basis functions can again be obtained by the product of univariate orthogonal polynomials employing the optimal univariate polynomial at each random dimension. This approach requires the input uncertainties to be independent standard random variables, which also allows the calculation of the multivariate weight functions by the product of univariate weight functions associated with the probability distribution at each random dimension. Detailed information on polynomial chaos expansions can be found in Walters and Huyse [53], Najm [54], and Hosder and Walters [55].

The objective of the stochastic methods based on polynomial chaos is to determine the coefficient of each term ($\alpha_j(\mathbf{x}, t)$ $(j = 0, 1, \ldots, P)$) in the polynomial expansion given by Eq. (10). The statistics of the response can then be calculated using the coefficients and the orthogonality of basis functions. The mean of the random solution is given by

$$\mu_R = E\big[R(\boldsymbol{\xi})\big] = \int_{\Omega} R(\boldsymbol{\xi}) \rho(\boldsymbol{\xi})\, d\boldsymbol{\xi} = \alpha_0, \tag{12}$$

which indicates that the zeroth mode of the expansion corresponds to the expected value or the mean of $R(\boldsymbol{\xi})$. Similarly, the variance of the distribution can be obtained

as

$$\sigma_R^2 = \mathrm{Var}\big[R(\boldsymbol{\xi})\big] = \int_{\Omega} \big(R(\boldsymbol{\xi}) - \mu_R\big)^2 \rho(\boldsymbol{\xi})\, d\boldsymbol{\xi} \tag{13}$$

or

$$\sigma_R^2 = \sum_{j=1}^{P} \big[\alpha_j^2 \langle \Psi_j^2 \rangle\big]. \tag{14}$$

In the above equations, we have used the fact that $\langle \Psi_j \rangle = 0$ for $j > 0$ and $\langle \Psi_i \Psi_j \rangle = \langle \Psi_j^2 \rangle \delta_{ij}$, where the inner product expression $\langle \cdots \rangle$ represents

$$\big\langle f(\boldsymbol{\xi}) g(\boldsymbol{\xi}) \big\rangle_{\boldsymbol{\xi}} = \int_{\Omega} f(\boldsymbol{\xi}) g(\boldsymbol{\xi}) \rho(\boldsymbol{\xi})\, d\boldsymbol{\xi} \tag{15}$$

written in terms of two generic functions $f(\boldsymbol{\xi})$ and $g(\boldsymbol{\xi})$ in the support region Ω of $\boldsymbol{\xi}$ with $\rho(\boldsymbol{\xi})$ as the weight function.

To model the uncertainty propagation in computational simulations via polynomial chaos with the intrusive approach, all dependent variables and random parameters in the governing equations are replaced with their polynomial chaos expansions. Taking the inner product of the equations (or projecting each equation onto the jth basis) yields $P+1$ times the number of deterministic equations which can be solved by the same numerical methods applied to the original deterministic system. Although straightforward in theory, an intrusive formulation for complex problems can be relatively difficult, expensive, and time-consuming to implement. To overcome these inconveniences associated with the intrusive approach, non-intrusive polynomial chaos formulations have been considered for uncertainty propagation.

Point-Collocation NIPC The point-collocation NIPC method starts with replacing the uncertain variables of interest with their polynomial expansions given by Eq. (10). Then, $N_t = P + 1$ vectors ($\boldsymbol{\xi}_j = \{\xi_1, \xi_2, \ldots, \xi_n\}_j$, $j = 0, 1, \ldots, P$) are chosen in random space for a given PC expansion with $P + 1$ modes, and the deterministic code is evaluated at these points. With the left-hand side of Eq. (10) known from the solutions of deterministic evaluations at the chosen random points, a linear system of equations can be obtained:

$$\begin{pmatrix} R(\overrightarrow{\xi_0}) \\ R(\overrightarrow{\xi_1}) \\ \vdots \\ R(\overrightarrow{\xi_P}) \end{pmatrix} = \begin{pmatrix} \Psi_0(\overrightarrow{\xi_0}) & \Psi_1(\overrightarrow{\xi_0}) & \cdots & \Psi_P(\overrightarrow{\xi_0}) \\ \Psi_0(\overrightarrow{\xi_1}) & \Psi_1(\overrightarrow{\xi_1}) & \cdots & \Psi_P(\overrightarrow{\xi_1}) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_0(\overrightarrow{\xi_P}) & \Psi_1(\overrightarrow{\xi_P}) & \cdots & \Psi_P(\overrightarrow{\xi_P}) \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_0 \end{pmatrix}. \tag{16}$$

The coefficients (α_j) of the stochastic expansion are obtained by solving the linear system of equations given above. The solution of the linear problem given by Eq. (16) requires N_t deterministic function evaluations. If more than N_t samples are chosen, then the overdetermined system of equations can be solved using the

least squares approach. Hosder et al. [56] investigated this option on model stochastic problems by increasing the number of collocation points in a systematic way through the introduction of an oversampling ratio (*OSR*) defined as the number of samples divided by N_t. Based on a study of different model problems, they suggested an effective *OSR* of 2.0. Point-collocation NIPC has the advantage of flexibility on the selection of collocation points. With the proper selection of collocation points, it has been shown that point-collocation NIPC can produce highly accurate stochastic response surfaces with computational efficiency [56]. In the model problems considered in this study, we use Latin hypercube sampling with an oversampling ratio of 1 or 2 to choose the collocation points. The number of response evaluations will be $OSR \times N_t$ when point-collocation NIPC is used to construct the stochastic response surface.

Quadrature-Based NIPC With the quadrature-based NIPC method, stochastic expansion coefficients α_j (see Hosder and Walters [55] for details) can be obtained by using the equation

$$\alpha_j = \frac{\langle R, \Psi_j(\boldsymbol{\xi}) \rangle}{\langle \Psi_j^2(\boldsymbol{\xi}) \rangle} = \frac{1}{\langle \Psi_j^2(\boldsymbol{\xi}) \rangle} \int_{\Omega} R \Psi_j(\boldsymbol{\xi}) \rho(\vec{\xi}) \, d\vec{\xi} \,. \tag{17}$$

Since the denominator $\langle \Psi_j^2 \rangle$ in Eq. (17) can be computed analytically for multivariate orthogonal polynomials, the main purpose is to compute the coefficients by estimating the numerator $\langle R, \Psi_j \rangle$ in Eq. (17). In the quadrature-based NIPC method, the approximation of multidimensional integrals can be achieved by applying a tensor product of one-dimensional quadrature rules. The Gaussian quadrature points are precisely the roots of the orthogonal polynomial on the same interval (the support region of the uncertain variable $\boldsymbol{\xi}$), and a weighting function is associated with the given uncertainty distribution (i.e., Gauss–Legendre and Gauss–Hermite quadrature for expansions on uniform and normal random variables, respectively). For one-dimensional integrals, if the polynomial chaos expansion degree is p, then the minimum Gaussian points required for the exact estimation of the integral will be $p + 1$ (with the assumption that the response R can be represented exactly with a polynomial expansion of p), since the p-point Gaussian quadrature rule will yield an exact result for polynomials of degree $2p - 1$ or less, and the polynomial degree of the product of function estimation and the basis polynomials in the numerator in Eq. (17) will be $2p$. Therefore, the number of response evaluations will be $(p + 1)^n$ when quadrature-based NIPC is used to construct the response surface as a function of n expansion variables. For multidimensional problems with a large number of uncertain variables, the computational cost of this method will be significant due to its exponential growth with the number of random dimensions. For these cases, an alternative approach for more efficient evaluation of the multidimensional integrals will be to use sparse tensor product spaces instead of full tensor products of Gauss quadrature points to cover the multidimensional random space (see Eldred et al. [27, 44] for details).

2.3 Utilization of Stochastic Expansions for Robust Design

The methodology described in this chapter employs the stochastic response surfaces obtained with NIPC methods described above. While constructing the stochastic response surfaces, a combined expansion approach is utilized, which expands the polynomials as a function of uncertain design variables and parameters (aleatory and epistemic), as well as the deterministic design variables. We will describe this approach and robust optimization with stochastic expansions in the following sections.

2.3.1 Formulation with Combined Expansions

With the introduction of deterministic design variables ($\mathbf{X}_d$), design variables with epistemic uncertainty ($\mathbf{X}_e$), parameters with epistemic uncertainty ($\mathbf{P}_e$), design variables with aleatory uncertainty ($\mathbf{X}_a$), and parameters with aleatory uncertainty ($\mathbf{P}_a$), a combined stochastic expansion of R is written as

$$R\big(\mathbf{X}_a(\boldsymbol{\xi}_{xa}), \mathbf{P}_a(\boldsymbol{\xi}_{pa}), \mathbf{X}_e(\boldsymbol{\xi}_{xe}), \mathbf{P}_e(\boldsymbol{\xi}_{pe}), \mathbf{X}_d(\boldsymbol{\xi}_d)\big) = \sum_{j=0}^{P} \alpha_j \Psi_j(\boldsymbol{\xi}_{xa}, \boldsymbol{\xi}_{pa}, \boldsymbol{\xi}_{xe}, \boldsymbol{\xi}_{pe}, \boldsymbol{\xi}_d). \tag{18}$$

In this approach, multidimensional basis functions Ψ_j are derived from the tensor product of one-dimensional optimum basis functions for the aleatory uncertain parameters and design variables ($\mathbf{P}_a$ and $\mathbf{X}_a$) selected based on their input probability distributions (e.g., Hermite polynomials for normal uncertain variables, the Legendre polynomials for the epistemic uncertain parameters and design variables ($\mathbf{P}_e$ and $\mathbf{X}_e$), and the Legendre polynomials for the deterministic design variables ($\mathbf{X}_d$)). The selection of the Legendre polynomials as basis functions for the epistemic uncertainties and the design variables is due to their bounded nature ($\mathbf{P}_e^L \le \mathbf{P}_e \le \mathbf{P}_e^U$, $\mathbf{X}_e^L \le \mathbf{X}_e \le \mathbf{X}_e^U$, and $\mathbf{X}_d^L \le \mathbf{X}_d \le \mathbf{X}_d^U$) and should not be interpreted as a probability assignment to these variables. In Eq. (18), $\boldsymbol{\xi}_{xa}$ and $\boldsymbol{\xi}_{pa}$ correspond to standard aleatory random variable vectors associated with $\mathbf{X}_a$ and $\mathbf{P}_a$, whereas $\boldsymbol{\xi}_{xe}$, $\boldsymbol{\xi}_{pe}$, and $\boldsymbol{\xi}_d$ are the standard variables in the interval $[-1, 1]$, which are mapped from the associated intervals of $\mathbf{X}_e$, $\mathbf{P}_e$, and $\mathbf{X}_d$ via

$$\boldsymbol{\xi}_{xe} = \left(\mathbf{X}_e - \left(\frac{\mathbf{X}_e^L + \mathbf{X}_e^U}{2}\right)\right) \Big/ \left(\frac{\mathbf{X}_e^U - \mathbf{X}_e^L}{2}\right), \tag{19}$$

$$\boldsymbol{\xi}_{pe} = \left(\mathbf{P}_e - \left(\frac{\mathbf{P}_e^L + \mathbf{P}_e^U}{2}\right)\right) \Big/ \left(\frac{\mathbf{P}_e^U - \mathbf{P}_e^L}{2}\right), \tag{20}$$

$$\boldsymbol{\xi}_d = \left(\mathbf{X}_d - \left(\frac{\mathbf{X}_d^L + \mathbf{X}_d^U}{2}\right)\right) \Big/ \left(\frac{\mathbf{X}_d^U - \mathbf{X}_d^L}{2}\right). \tag{21}$$

Using the combined expansion given in Eq. (18), the mean and the variance of the response are obtained by evaluating the expectations over the aleatory uncertain variables ($\boldsymbol{\xi}_{xa}$ and $\boldsymbol{\xi}_{pa}$), which will be functions of standard epistemic design variables ($\boldsymbol{\xi}_{xe}$), standard epistemic parameters ($\boldsymbol{\xi}_{pe}$), and standard deterministic design variables ($\boldsymbol{\xi}_{d}$):

$$\mu_R(\boldsymbol{\xi}_{xe}, \boldsymbol{\xi}_{pe}, \boldsymbol{\xi}_{d}) = \sum_{j=0}^{P} \alpha_j \left\langle \Psi_j(\boldsymbol{\xi}_{xa}, \boldsymbol{\xi}_{pa}, \boldsymbol{\xi}_{xe}, \boldsymbol{\xi}_{pe}, \boldsymbol{\xi}_{d}) \right\rangle_{\boldsymbol{\xi}_{xa}, \boldsymbol{\xi}_{pa}}, \tag{22}$$

$$\sigma_R^2(\boldsymbol{\xi}_{xe}, \boldsymbol{\xi}_{pe}, \boldsymbol{\xi}_{d}) = \left(\sum_{j=0}^{P} \sum_{k=0}^{P} a_j a_k \langle \Psi_j \Psi_k \rangle_{\boldsymbol{\xi}_{xa}, \boldsymbol{\xi}_{pa}} \right) - \mu_R^2. \tag{23}$$

2.3.2 Robust Design Based on Stochastic Expansions

The flowchart of robust optimization under mixed uncertainties based on combined stochastic expansions is shown in Fig. 5. From Eqs. (22) and (23) it can be clearly seen that the mean and standard deviation of the response R (i.e., objective function F or constraint g_i) at a given design point are characterized by two bounds due to epistemic uncertainties with specified interval bounds $[\mathbf{X}_e^L, \mathbf{X}_e^U]$ and $[\mathbf{P}_e^L, \mathbf{P}_e^U]$. In other words, the mean and standard deviation of the output (response) will also be bounded by its maximum and minimum values. In our approach, we calculate $\mu_R^{\max}$, $\mu_R^{\min}$, $\sigma_R^{\max}$, and $\sigma_R^{\min}$ at a given design point through optimization using the analytical expressions of response statistics obtained with Eq. (22) and Eq. (23). Then, these values are used in the robust optimization formulation given by Eq. (9), which is performed with the sequential quadratic programing (SQP) method [57]. The whole procedure is repeated until the convergence is achieved. Note that when at least one design variable is uncertain (aleatory or epistemic), the stochastic response surfaces for the objective function and the constraints (if necessary) have to be reconstructed at each optimization iteration, since the uncertain design variables and the associated statistics are updated at each iteration, changing the bounds on which the response surfaces are created. On the other hand, if all design variables are deterministic and the uncertainties are associated with the problem parameters, only a single stochastic response surface for the objective function and a single response surface for each constraint function have to be constructed, since the bounds on the statistics of uncertain parameters and the bounds on the design variables are fixed and do not vary during the entire optimization process.

3 Model Problem: Robust Design of a Beam

In this model problem, which includes uncertainties in both design variables and parameters, we consider the robust design of a cantilever beam shown in Fig. 6 with

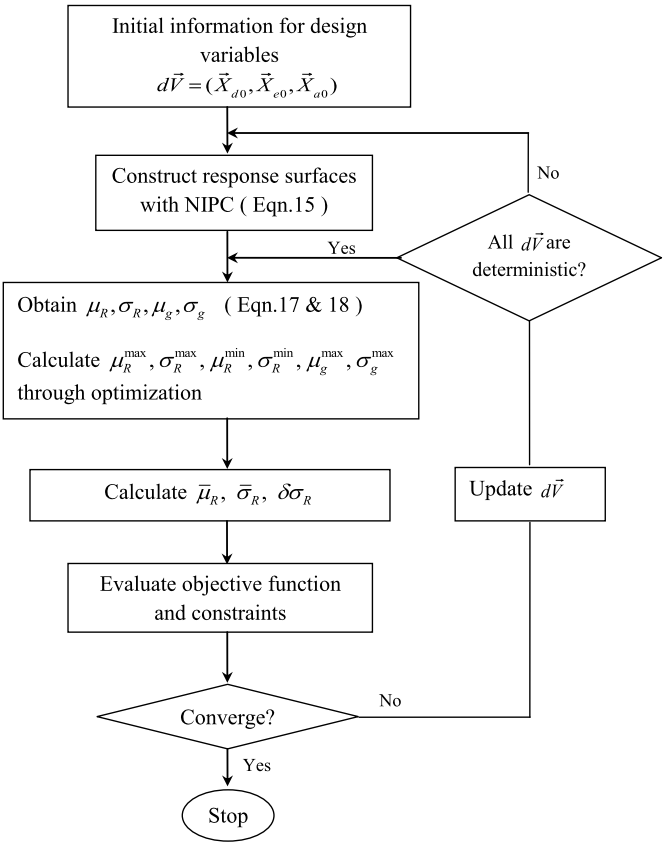


Fig. 5 Flowchart of the robust optimization process under mixed uncertainties with combined stochastic expansions

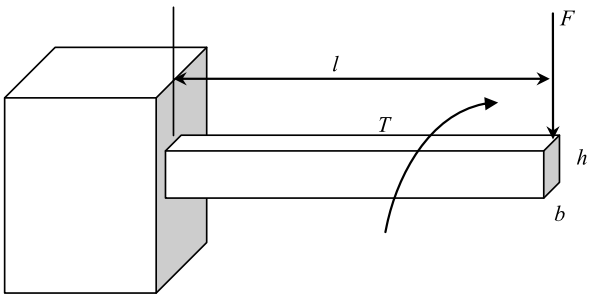


Fig. 6 Schematic of the beam design problem

length l, width b, and height h. The beam is subjected to a torque T and an external force F acting normal to the horizontal axis of the beam at its free end. The objective

Table 1 Design variables (DV) and parameters (P) with epistemic uncertainty (beam design model problem)

DV / P	lower limit	upper limit
$\overline{l}$ (DV)	$\overline{l} - 0.1\overline{l}$	$\overline{l} + 0.1\overline{l}$
F (P)	270 lb	330 lb

Table 2 Design variables (DV) and parameters (P) with aleatory uncertainty (beam design model problem)

DV / P	Mean	Standard deviation	Distribution
h (DV)	μ_h	1 %μ_h	Normal
b (DV)	μ_b	1 %μ_b	Normal
S (P)	100 kpsi	10 kpsi	Normal
T (P)	450 lb-in	50 lb-in	Normal

is to reduce the volume ($V = lbh$) of the beam while satisfying a stress constraint given by

$$g = \sqrt{\left(\frac{6FL}{bh^2}\right)^2 + 3\left[\frac{T}{b^2h}\left(3 + \frac{1.8b}{h}\right)\right]^2} - S \leqslant 0. \tag{24}$$

This equation represents the difference between the maximum equivalent stress of the beam and the yield strength S, which must be less than or equal to zero for a safe design. In this design problem, the external force F is considered as a parameter with epistemic uncertainty and the length of the beam l is treated as an epistemic design variable (Table 1). The external torque T and yield strength S are treated as parameters with aleatory uncertainty, whereas the width b and the height h of the beam are modeled as aleatory design variables with statistics given in Table 2. To ensure robustness of the design under epistemic and aleatory uncertainties, the objective function, which is the weighted sum of the average mean of the volume ($\overline{\mu}_V$), the average standard deviation of the volume ($\overline{\sigma}_V$), and the difference between the maximum and minimum standard deviation of the volume ($\delta\sigma_V$) should be minimized. The inequality constraint given by Eq. (24) should be satisfied at the worst case with a specified β value of 3. With the addition of the limits for the design variables, the robust design formulation for this problem is given as:

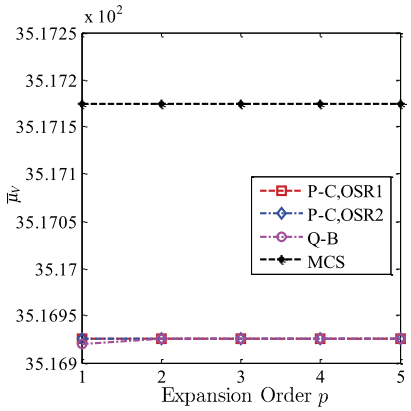
$$\begin{aligned} \min_d \quad & w_1\overline{\mu}_V + w_2\overline{\sigma}_V + w_3\delta\sigma_V \\ \text{s.t.} \quad & \mu_g^{\max} + \beta\sigma_g^{\max} \leq 0 \\ & 0.1 \leq \mu_h \leq 0.8 \\ & 0.1 \leq \mu_b \leq 0.4 \\ & 2 \leq \overline{l} \leq 20. \end{aligned} \tag{25}$$

Considering the magnitude of $\overline{\mu}_V$, $\overline{\sigma}_V$, and $\delta\sigma_V$, the weights in the multi-objective function are chosen as $w_1 = 1$, $w_2 = 100$, and $w_3 = 500$ to ensure equal contribu-

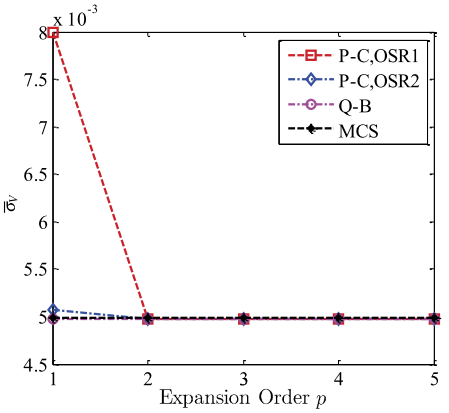
tions to the objective function from each term (i.e., scaling them to approximately the same order of magnitude).

For this problem, besides the stochastic response surface-based approach, robust optimization was also performed with double-loop Monte Carlo sampling (MCS), which was the approach used by Du et al. [48] to propagate the mixed uncertainties and obtain the maximum and minimum value of the response statistics used in the robust optimization formulation. After performing a convergence study based on the inner and outer loop samples, the desired accuracy with the double-loop MCS approach for the robustness measures was obtained with 500 epistemic variable samples in the outer loop and 10^5 aleatory variable samples in the inner loop. The convergence of the performance and robustness measures used in the objective function ($\bar{\mu}_V$, $\bar{\sigma}_V$, and $\delta\sigma_V$) obtained with quadrature-based and point-collocation NIPC with $OSR = 1$ and $OSR = 2$ was studied for different polynomial expansion orders at the optimum design point obtained with the Monte Carlo approach (Fig. 7). For the same robustness measures, the error values relative to the Monte Carlo results at each polynomial order are shown in Fig. 8. It is evident that the convergence is rapid for $\overline{\mu}_V$ and is achieved by the first-order expansion for all NIPC methods. The convergences for $\overline{\sigma}_V$ and $\delta\sigma_V$ are obtained at the second-order expansion. From Fig. 8, it can be seen that quadrature-based NIPC is more accurate than the point-collocation-based approach in terms of the error levels (especially for $\delta\sigma_V$) observed at the second-order expansion.

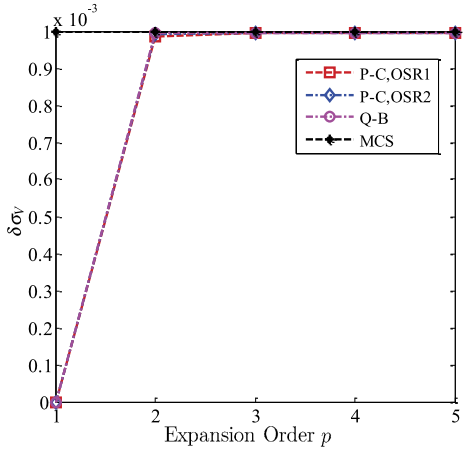
Based on the convergence results, the robust optimization was performed with stochastic response surfaces representing the objective function and the inequality constraint obtained with the NIPC approach utilizing a second-order polynomial expansion over aleatory and epistemic design variables and parameters. The two NIPC methods, point-collocation and quadrature-based, were implemented to obtain the stochastic response surfaces for comparison. The point-collocation method was performed with an OSR of 1 and 2. The number of original function evaluations required to construct a single response surface was 28 for the point-collocation method with $OSR = 1$ and 56 with $OSR = 2$. The quadrature-based method required 729 function evaluations. For this case, two response surfaces were constructed at each optimization iteration, one for the objective function and the other for the inequality constraint. The optimum design variable values obtained with the stochastic response surface based optimization again compare well with the result of the approach utilizing the double-loop MCS (Table 3) at a significantly reduced cost in terms of the total number of function evaluations, which include both the objective and constraint functions. Point-collocation NIPC is more efficient than quadrature-based NIPC for this problem, since the number of expansion variables ($n = 6$) is greater than the number of variables in the previous model problem ($n = 3$) and the computational cost of quadrature-based NIPC increases exponentially with the number of expansion variables for a given polynomial degree. An alternative approach to reduce the computational expense of the quadrature-based approach will be to implement sparse grid and cubature techniques, which may improve the computational efficiency significantly while retaining the accuracy of the original tensor product quadrature method. Table 4 presents the average mean, average standard



(a) Average mean of beam volume, $\bar{\mu}_V$



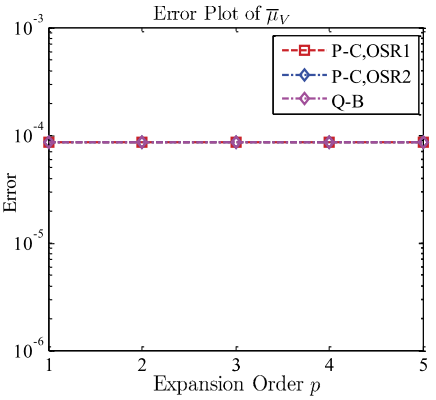
(b) Average standard deviation of beam volume, $\bar{\sigma}_V$



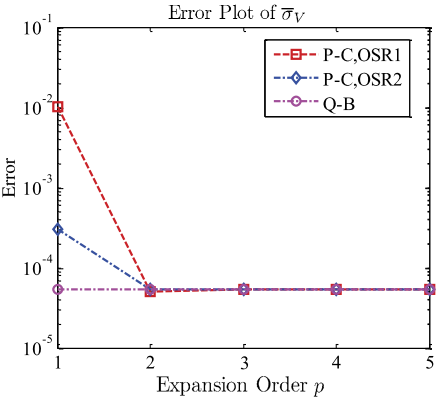
(c) Difference of standard deviation of beam volume, $\delta\sigma_V$

Fig. 7 Convergence of NIPC results as a function of expansion order for beam design model problem

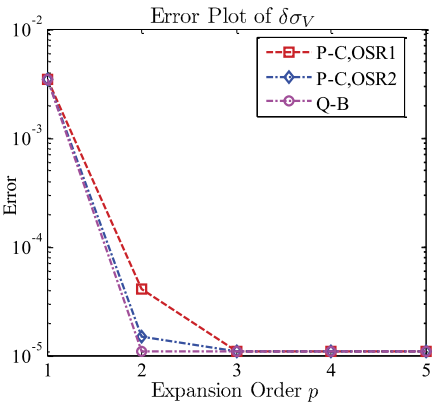
deviation, and the standard deviation difference of the beam volume at the optimum design point; these are approximately the same for all the methods. The convergence histories of these terms are given in Fig. 9 for the optimization process with stochastic expansions. As can be seen from this figure, all three quantities are minimized simultaneously and converge to the same final values, which validates the described stochastic response surface based robust optimization approach. Another important observation made from this figure is that the quadrature-based approach seems to converge to the optimum robust design in terms of all measures at a lower number of iterations and in a more stable manner compared to the point-collocation-based



(a) Error of average mean of beam volume, $\bar{\mu}_V$



(b) Error of average standard deviation of beam volume, $\bar{\sigma}_V$



(c) Error of difference of standard deviation of beam volume, $\delta\sigma_V$

Fig. 8 Error convergence of NIPC results as a function of expansion order for beam design model problem

Table 3 Optimum design results of the beam problem (RS: response surface, Q-B: quadrature-based, P-C: point-collocation, MCS: Monte Carlo sampling, FE: function evaluations)

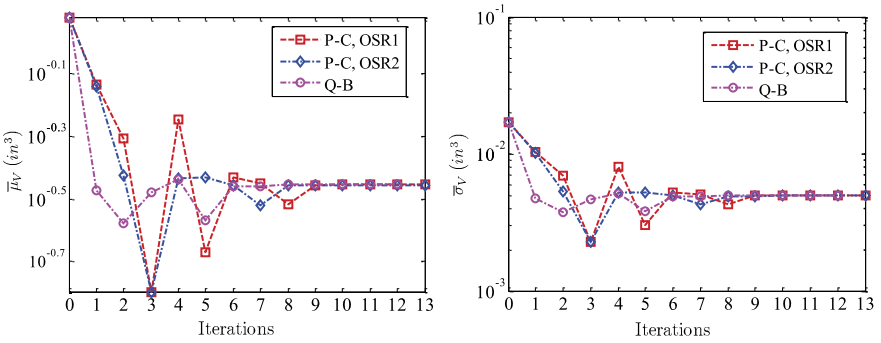
Method	$\{\mu_h, \mu_b, \bar{l}\}$ (in)	Total # of FE
MCS	{0.548, 0.327, 2.0}	23.5×10^8
Q-B	{0.542, 0.323, 2.0}	96,228
P-C, OSR = 1	{0.543, 0.324, 2.0}	3,696
P-C, OSR = 2	{0.542, 0.323, 2.0}	6,720

methods, especially the one with $OSR = 1$. This emphasizes another aspect of the importance of the accuracy of the stochastic response surfaces used in the robust

Table 4 Robustness assessment of the beam problem

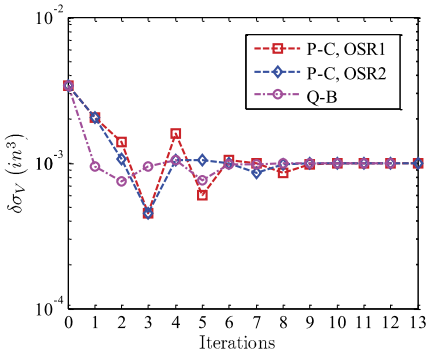
Method	$\overline{\mu}_V$	$\overline{\sigma}_V$	$\delta\sigma_V$
MCS	3.55×10^{-1}	5.06×10^{-3}	1.01×10^{-3}
Q-B	3.50×10^{-1}	4.96×10^{-3}	9.91×10^{-4}
P-C, OSR = 1	3.52×10^{-1}	4.97×10^{-3}	9.86×10^{-4}
P-C, OSR = 2	3.50×10^{-1}	4.94×10^{-3}	9.88×10^{-4}

optimization approach in terms of the number of iterations to converge, which may influence the computational efficiency of the overall stochastic optimization process.



(a) Average mean of beam volume, $\bar{\mu}_V$

(b) Average standard deviation of beam volume, $\bar{\sigma}_V$



(c) Difference of standard deviation of beam volume, $\delta\sigma_V$

Fig. 9 The convergence history of average mean, average standard deviation, and the standard deviation difference of the beam volume for the optimization process with stochastic expansions

4 Application Example: Robust Airfoil Design

In this section, a computationally efficient approach for robust aerodynamic optimization under aleatory (inherent) and epistemic (model-form) uncertainties using stochastic expansions based on the point-collocation NIPC method is described. The deterministic CFD simulation model used in the optimization studies is described. Then the robust aerodynamic optimization formulations are given, followed by a discussion on the utilization of stochastic expansions in robust optimization. The results of two case studies are presented.

4.1 Computational Fluid Dynamics and Airfoil Shape Model

This section describes the elements of the CFD model, including the governing equations, numerical solution of the governing fluid flow equations (flow solver), the airfoil shape model, and meshing of the solution domain.

4.1.1 Governing Equations

The flow is assumed to be steady, two dimensional, compressible, and turbulent. The steady Reynolds-averaged Navier–Stokes (RANS) equations are taken as the governing fluid flow equations. The fluid medium is air, assumed to be an ideal gas, with the laminar dynamic viscosity (μ) described by Sutherland's formula (see, e.g., Ref. [58]). For modeling the turbulent kinematic eddy viscosity (ν_t), we use the turbulence model by Spalart and Allmaras [59]. The Spalart–Allmaras model, designed specifically for aerodynamic wall-bounded flows, is a one-equation model that solves a single conservation partial differential equation for the turbulent viscosity. This conservation equation contains convective and diffusive transport terms, as well as expressions for the production and dissipation of ν_t. The Spalart–Allmaras model is economical and accurate for attached wall-bounded flows and flows with mild separation and recirculation. However, the model may not be accurate for massively separated flows, free shear flows, and decaying turbulence. As described later, the turbulent viscosity is multiplied by a factor k to introduce the epistemic uncertainty in our robust optimization under mixed uncertainties problem. This is implemented in the solution through a user-defined function (UDF) which is dynamically loaded with the flow solver (described below) for each CFD simulation. The whole procedure is executed automatically through scripts.

4.1.2 Flow Solver

The flow solver is of implicit density-based formulation, and the fluxes are calculated by an upwind-biased second-order spatially accurate Roe flux scheme. Asymptotic convergence to a steady state solution is obtained for each case. Automatic solution steering is employed to gradually ramp up the Courant number and accelerate

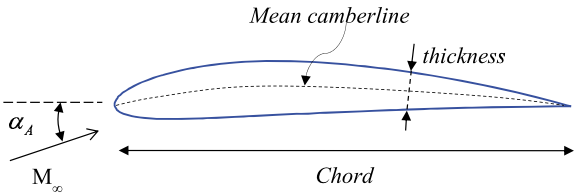


Fig. 10 A typical NACA four-digit airfoil section is shown. The free-stream flow is at Mach number M_∞, at an angle of attack α_A relative to the chord axis

convergence. Full multigrid initialization is used to get a good starting point. Numerical fluid flow simulations are performed using the computer code FLUENT [60].

The iterative convergence of each solution is examined by monitoring the overall residual, which is the sum (over all the cells in the computational domain) of the L^2 norm of all the governing equations solved in each cell. In addition to this, the lift and drag forces are monitored for convergence. The solution convergence criterion for the CFD runs is the one that occurs first of the following: a maximum residual of 10^{-6}, or a maximum number of iterations of 1,000.

4.1.3 Airfoil Geometry

In this work, we use the National Advisory Committee for Aeronautics (NACA) airfoil shapes. In particular, we use the NACA four-digit airfoil parameterization method, where the airfoil shape is defined by three parameters: c (the maximum ordinate of the mean camber line as a fraction of the chord), l_c (the chordwise position of the maximum ordinate as a fraction of the chord), and t (the thickness-to-chord ratio). The airfoils are denoted by NACA $mpxx$, where xx represents $(100 \times t)$, m is equal to $(100 \times c)$, and p is $(10 \times l_c)$. The shapes are constructed using two polynomials, one for the thickness distribution and the other for the mean camber line. The full details of the NACA four-digit parameterization are given in Abbott and von Doenhoff [61]. A typical NACA four-digit airfoil section is shown in Fig. 10.

4.1.4 Grid Generation

The solution domain boundaries are placed at 25 chord lengths in front of the airfoil, 50 chord lengths behind it, and 25 chord lengths above and below it. The computational meshes are of structured curvilinear body-fitted C-topology with elements clustering around the airfoil and growing in size with distance from the airfoil surface. The non-dimensional normal distance (y^+) from the wall to the first grid point is roughly one. The free-stream Mach number, angle of attack, static pressure, and the turbulent viscosity ratio are prescribed at the far-field boundary. An example of a computational grid is shown in Fig. 11. The computer code ICEM CFD [62] is used for the mesh generation.

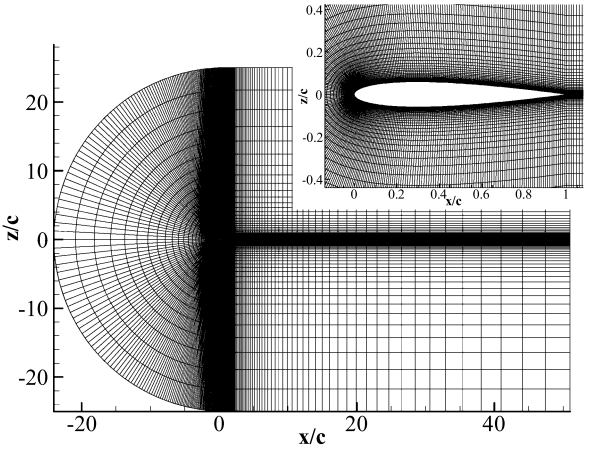


Fig. 11 An example computational grid for the NACA 0012 airfoil

4.2 Robust Airfoil Optimization Formulation

Here, the details are given for the robust aerodynamic optimization with stochastic expansions for two cases: (1) optimization under pure aleatory uncertainty and (2) optimization under mixed (aleatory and epistemic) uncertainty.

4.2.1 Optimization Under Pure Aleatory Uncertainty

The robust airfoil optimization under pure aleatory uncertainty is formulated as

$$\begin{aligned} \min \quad & \mu_{C_d} + \sigma_{C_d} \\ \text{subject to} \quad & \mu_{C_L} \geq C_L^* \\ & 0.0 \leq c \leq 0.05 \\ & 0.3 \leq l_c \leq 0.7 \\ & 0.08 \leq t \leq 0.14, \end{aligned} \tag{26}$$

where $C_d = C_d(\mathbf{X}_d, \mathbf{P}_a)$ is the profile drag coefficient, which is a function of the deterministic design variable vector $\mathbf{X}_d$ and aleatory input uncertainty vector $\mathbf{P}_a$. Similarly, the lift coefficient is $C_L = C_L(\mathbf{X}_d, \mathbf{P}_a)$. In our current optimization study, we use the deterministic design variable vector $\mathbf{X}_d = \{c, l_c, t\}$ to control our airfoil shape. Note that this vector can contain the control points as the design variables when the airfoil shape is parameterized with different spline fitting techniques. The free-stream Mach number (M_∞) is treated as aleatory (inherent) input uncertainty (i.e., $\mathbf{P}_a = \{M_\infty\}$) and represented as a uniform random variable with bounds ($0.7 \leq M_\infty \leq 0.8$). As can be seen from the Mach number range, we focus on the

transonic flow regime, where the drag coefficient is very sensitive to the changes in the Mach number due to the lambda shock originating on the top surface of the airfoil. This minimization is subject to satisfying a desired profile lift coefficient (C_L^*) value or higher by the mean value of the lift coefficient adjusted by changing the angle of attack for a given design variable and uncertain variable vector value. The optimization also includes geometric constraints for the profile shape, which bound the thickness, maximum camber, and the maximum camber location (note that l_c is taken as zero when $c = 0.0$). The drag coefficient and other aerodynamics characteristics of the airfoil for a given design variable vector and aleatory uncertain variable value are obtained from the CFD simulations that solve steady, two-dimensional, Reynolds-averaged Navier–Stokes equations with Spalart–Allmaras turbulence model.

4.2.2 Optimization Under Mixed Uncertainty

The robust airfoil optimization under mixed (aleatory and epistemic) uncertainties is formulated as

$$\begin{aligned} \min \quad & w_1 \overline{\mu}_{C_d} + w_2 \overline{\sigma}_{C_d} + w_3 \delta\sigma_{C_d} \\ \text{subject to} \quad & \mu_{C_L}^{\min} \geq C_L^* \\ & 0.0 \leq c \leq 0.05 \\ & 0.3 \leq l_c \leq 0.7 \\ & 0.08 \leq t \leq 0.14, \end{aligned} \tag{27}$$

where the profile drag coefficient $C_d(\mathbf{X}_d, \mathbf{P}_a, \mathbf{P}_e)$ is now a function of the deterministic design variable vector $\mathbf{X}_d$, aleatory input uncertainty vector $\mathbf{P}_a$, and the epistemic input uncertainty vector $\mathbf{P}_e$. Similarly, the lift coefficient, $C_L(\mathbf{X}_d, \mathbf{P}_a, \mathbf{P}_e)$, is now a function of the same variables. In the multi-objective function w_1, w_2, w_3 are the weights whose sum is equal to 1.0. In this study, we have used equal weights; however, one can choose different weights depending on the emphasis on each term.

In this optimization problem, we again consider airfoil shape parameters as our deterministic design variable vector ($\mathbf{X}_d = \{c, l_c, t\}$ and the free-stream Mach number as the aleatory (inherent) input uncertainty ($\mathbf{P}_a = \{M_\infty\}$) with bounds ($0.7 \leq M_\infty \leq 0.8$). The kinematic eddy viscosity (ν_t) obtained from the Spalart–Allmaras turbulence model used in RANS simulations is modeled as an epistemic (model-form) input uncertainty (i.e., $\mathbf{P}_e = \{\nu_t\}$) through the introduction of a factor k as shown below:

$$\nu_t = k\nu_{t_{\text{SA}}}, \tag{28}$$

where $\nu_{t_{\text{SA}}}$ is the turbulent viscosity originally obtained with the Spalart–Allmaras model. The range of this factor k is chosen between 0.5 and 2.0 to mimic the model-form uncertainty due to the use of different turbulence models in RANS calculations. Figure 12 shows the pressure distributions of a NACA 2412 airfoil at

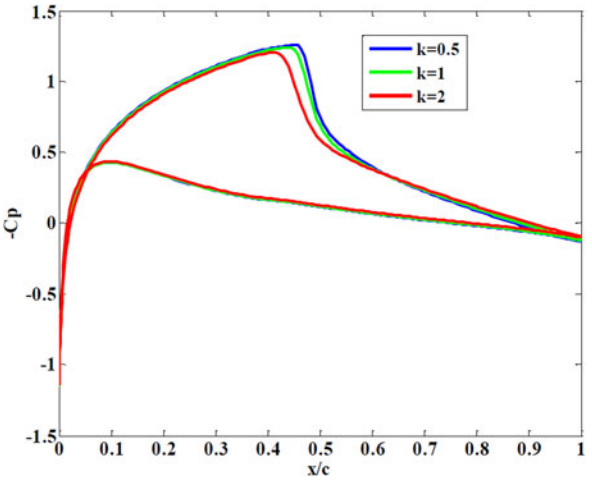


Fig. 12 The pressure distributions of NACA 2412 at $M_\infty = 0.75$, $\alpha_A = 1^\circ$

$M_\infty = 0.75$, $\alpha_A = 1^\circ$ with different k values. From this figure, it can be seen that the k factor and thus the turbulence model have a considerable effect on the pressure distribution, especially on the shock location.

This optimization is again subject to satisfying a desired profile lift coefficient (C_L^*) value or higher by the minimum of the mean value of the lift coefficient adjusted by changing the angle of attack for a given design variable and uncertain variable vector value. The optimization also includes geometric constraints for the profile shape, which bound the thickness, maximum camber, and the maximum camber location.

4.3 Utilization of Stochastic Response for Robust Optimization

Here, a stochastic response surface obtained with the point-collocation NIPC method is used for the propagation of aleatory and epistemic uncertainties due to its computational efficiency and accuracy as shown in the previous studies [43, 50]. When constructing the stochastic expansions, a combined expansion approach will be utilized, which will expand the polynomials as a function of both uncertain variables (aleatory and epistemic) and deterministic design variables.

Since the angle of attack has to be adjusted to satisfy the lift coefficient constraint in both optimization problems, we create three separate stochastic response surfaces with the point-collocation NIPC at three angles of attack ($\alpha_{A0} = 0.0^\circ$, $\alpha_{A1} = 1.0^\circ$, and $\alpha_{A2} = 2.0^\circ$). Then we apply Lagrange interpolation to create a composite response surface using these three response surfaces, which is continuous and quadratic in α_A between $\alpha_A = 0.0^\circ$ and $\alpha_A = 2.0^\circ$ for all design variables, aleatory uncertain variables (i.e., M_∞), and the epistemic uncertain variables (i.e., k). This composite response surface $\hat{R}$ (i.e., C_d or C_L), which is now a function of

α_A, $P_a(\boldsymbol{\xi}_{pa})$, $P_e(\boldsymbol{\xi}_{se})$, and $X_d(\boldsymbol{\xi}_d)$, can be written as

$$\hat{R}\big(\alpha_A, P_a(\boldsymbol{\xi}_{pa}), P_e(\boldsymbol{\xi}_{pe}), X_d(\boldsymbol{\xi}_d)\big) \cong \sum_{k=0}^{n_{\alpha_A}} R\big(P_a(\boldsymbol{\xi}_{pa}), P_e(\boldsymbol{\xi}_{pe}), X_d(\boldsymbol{\xi}_d)\big)_{\alpha_{Ak}} L_{n_{\alpha_A},k}(\alpha_A). \tag{29}$$

Based on this equation, we can slightly modify Eqs. (22) and (23) to calculate the mean and variance from $\hat{R}$:

$$\mu_{\hat{R}} = \mu_{\hat{R}}(\alpha_A, \boldsymbol{\xi}_{pe}, \boldsymbol{\xi}_d) = \sum_{k=0}^{n_{\alpha_A}} \sum_{j=0}^{P} \alpha_j(\alpha_{Ak}) L_{n_{\alpha_A},k}(\alpha_A) \big\langle \Psi_j(\boldsymbol{\xi}_{pa}, \boldsymbol{\xi}_{pe}, \boldsymbol{\xi}_d) \big\rangle_{\boldsymbol{\xi}_{pa}}, \tag{30}$$

$$\sigma^2_{\hat{R}} = \big\langle (\hat{R} - \hat{\mu_R})^2 \big\rangle_{\boldsymbol{\xi}_{pa}} = \left\{ \sum_{k=0}^{n_{\alpha_A}} \sum_{l=0}^{n_{\alpha_A}} \sum_{i=0}^{P} \sum_{j=0}^{P} L_{n_{\alpha_A},k}(\alpha_A) L_{n_{\alpha,l}}(\alpha_A) \alpha_i(\alpha_{Ak}) \alpha_j(\alpha_{Al}) \langle \Psi_i \Psi_j \rangle_{\boldsymbol{\xi}_{pa}} \right\} - \mu^2_{\hat{R}}, \tag{31}$$

where α_A is the angle of attack, $n_{\alpha_A} = 2$ is the degree of interpolation in α_A, and $L_{n_{\alpha,k}}$ is the Lagrange polynomial at α_{Ak} given by

$$L_{n_{\alpha_A},k}(\alpha_A) = \prod_{i=0, i\neq k}^{n_{\alpha_A}} \left(\frac{\alpha_A - \alpha_{Ai}}{\alpha_{Ak} - \alpha_{Ai}} \right). \tag{32}$$

Note that the introduction of α_A to the problem with the above approach indicates that it is considered as a deterministic design variable within $\hat{R}$. An alternative approach to involve α_A in the response surface would be to include it among the other deterministic design variables during the original construction.

The above formulations show that the mean and the standard deviation of response variables at a design point and angle of attack are characterized by two bounds due to epistemic uncertainties with specified interval bounds $[\mathbf{P}_e^L, \mathbf{P}_e^U]$. In other words, the mean and standard deviation of response R (i.e., C_d or C_L) at a design point and angle of attack will also be bounded by its maximum and minimum values. Once we create the composite stochastic response surface (Eq. (29)), we can use Eqs. (30) and (31) to calculate the mean and the standard deviation and use any standard optimization technique to determine $\mu_{c_d}^{\max}$, $\mu_{c_d}^{\min}$, $\sigma_{c_d}^{\max}$, and $\sigma_{c_d}^{\min}$ at a given design point and angle of attack.

With the combined expansion approach, it will be straightforward to calculate the total number of CFD simulations (N_{CFD}) required to create the composite response surface $\hat{R}$ that will be used in the entire optimization process:

$$N_{\text{CFD}} = OSR \times N_t \times (n_{\alpha_A} + 1), \tag{33}$$

where N_t is calculated from Eq. (11), and n_{α_A} is the degree of interpolation in α_A used in the creation of the composite response surface. It is important to note that N_{CFD} will be the computational cost of the described optimization approach, since once the response surface is created, the numerical evaluations at each optimization step will be computationally cheap due to the polynomial nature of the stochastic surrogate. Note that for a stochastic optimization problem with only a few design variables (i.e., $N_d \leq 3$), the combined expansion approach described above will be computationally very efficient, since a single response surface (a surrogate) is created which is a function of the design, aleatory, and epistemic uncertain variables. The optimization can be performed using this single response surface. On the other hand, in optimization problems with a large number of design variables, one can choose an alternative approach which is based on the expansion of the polynomial chaos surface only on the uncertain (aleatory and epistemic) variables. With this approach a separate stochastic response surface should be created at each design point, which will increase the computational cost; however, the accuracy of the response surface approximation will increase due to the reduction in the number of expansion variables.

4.4 Results and Discussion

The robust airfoil optimization approach with NIPC stochastic response surface is demonstrated here for two cases: (1) optimization under pure aleatory input uncertainty and (2) optimization under mixed (aleatory and epistemic) uncertainty.

4.4.1 Optimization Results for the Pure Aleatory Uncertainty Case

The free-stream Mach number is the only uncertain variable for this case and modeled with a uniform probability distribution between $M_\infty = 0.7$ and $M_\infty = 0.8$. The objective (Eq. (26)) is to reduce the mean and the standard deviation of the drag coefficient simultaneously to obtain an airfoil shape with minimum drag that is least sensitive to the change in Mach number in the specified range. Besides the side (geometric) constraints on the design variables, the minimization is performed such that the mean lift coefficient obtained with the optimum design is greater than or equal to 0.5. The stochastic response surfaces for the drag and the lift coefficients were created with the point-collocation NIPC method using a quadratic polynomial expansion with an OSR of 2 for four variables (three deterministic design variables and one uncertain variable). The total number of CFD evaluations required for this case was $N_{\text{CFD}} = 90$, as can be calculated using Eq. (33) along with Eq. (11). Note that this number is considerably low compared to the cost of alternative robust optimization formulations, which utilize Monte Carlo simulations for the calculation of the statistics. After the stochastic response surfaces for the drag and lift coefficients

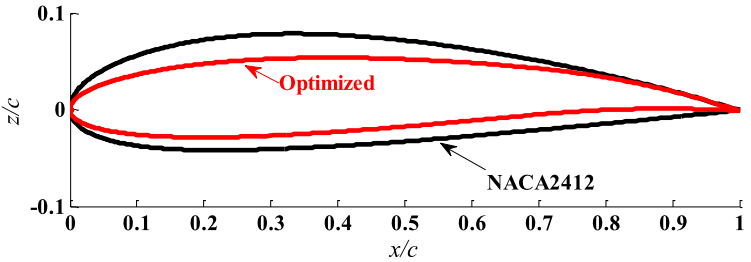


Fig. 13 NACA 2412 and the optimized airfoil shapes for the pure aleatory uncertainty case

Table 5 Optimization results for the pure aleatory uncertainty case

	Initial airfoil			Optimized airfoil		
	c	l_c	t	c	l_c	t
NACA 2412	0.020	0.40	0.120	0.0195	0.70	0.080
NACA 0012	0.0	0.0	0.120	0.0195	0.70	0.080

are created, the robust optimization is performed using the approach described in the previous section.

The robust optimization was performed starting from two different initial airfoil geometries (NACA 2412 and NACA 0012). As can be seen from Table 5, both optimization runs converged to the same optimum airfoil shape with $t = 0.08$, $c = 0.0195$, and $l_c = 0.7$ (Fig. 13). The optimum airfoil has the minimum thickness allowable and the camber is located as aft as possible toreduce the drag while satisfying the required C_L, which are typical characteristics of airfoils designed to operate at transonic speeds (e.g., supercritical airfoils). The camber value is the optimum to produce the required lift at an optimum angle of attack.

The pressure distributions of the NACA 2412 and optimum airfoils at $M_\infty = 0.75$ are shown in Fig. 14. From this figure we can see that, at $M_\infty = 0.75$, the NACA 2412 airfoil has a shock wave on the top surface, whereas no shock wave exists on the optimized airfoil, due to the increase in minimum suction pressure (i.e., the decrease of the maximum value of $-C_p$) and the reduction in the maximum velocity value on the top surface, giving a more flat pressure distribution. The aft camber compensates the lift that is lost in the suction region by loading the airfoil in the aft region.

The optimization history of the mean and the standard deviation of the drag coefficient is given in Fig. 15, which shows that both quantities are minimized simultaneously regardless of the initial airfoil chosen, which confirms the robust optimization approach used. This result is further verified by Fig. 16, which gives drag versus Mach number over the uncertain Mach number range for the NACA 2412 and optimized airfoil at a lift coefficient value of 0.5. As can be seen from this plot, the drag rise of NACA 2412 is significant, whereas the optimum airfoil maintains a low drag coefficient value over the uncertain Mach number range with no significant variation. The drag coefficient and L/D values for both airfoils are reported in Ta-

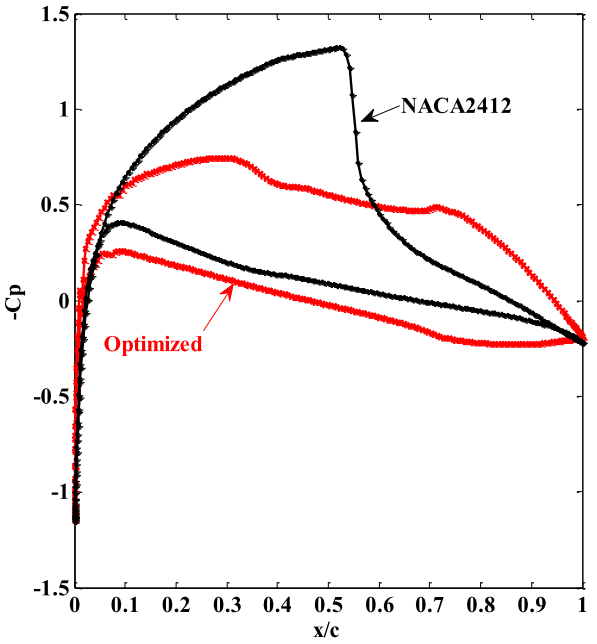


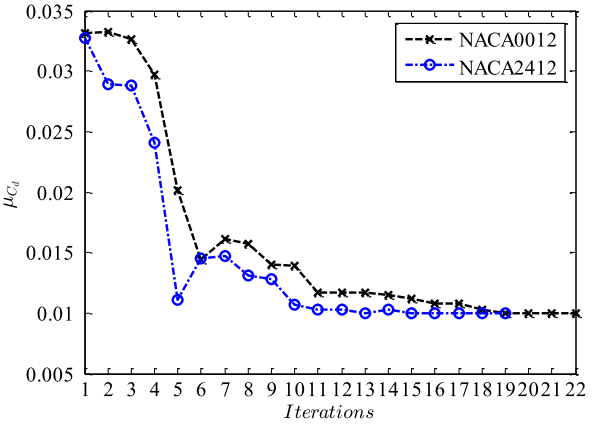
Fig. 14 The pressure distributions of NACA 2412 and optimum airfoil at $M_\infty = 0.75$ for the pure aleatory uncertainty case

ble 6, which quantifies the better aerodynamic performance of the optimum airfoil compared to NACA 2412.

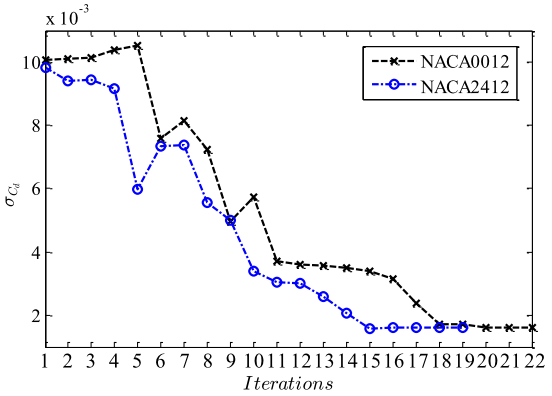
The drag characteristics of both airfoils can also be explained by examining the Mach number contours given in Fig. 17. As the Mach number increases, the shock wave on the top surface of the NACA 2412 airfoil gets stronger and eventually induces the boundary layer separation at a free-stream Mach number of 0.8, increasing the drag significantly. On the other hand, the delayed shock formation on the top surface of the optimum airfoil shape prevents a significant drag rise over the uncertain Mach number range considered. The pressure distributions of the NACA 2412 and optimum airfoils at $M_\infty = 0.7$, 0.75, and 0.8 are shown in Fig. 18. It can be seen that at $M_\infty = 0.7$, 0.75, the NACA 2412 airfoil has a shock wave on the top surface, whereas no shock wave exists on the optimized airfoil. At $M_\infty = 0.8$, the shock wave on the optimized airfoil is much weaker compared to the shock on the NACA 2412 airfoil.

4.4.2 Optimization Results for the Mixed Uncertainty Case

As described with the optimization formulation in Sect. 4.2.2, the free-stream Mach number is taken as the aleatory uncertain variable for this case and modeled with a uniform probability distribution between $M_\infty = 0.7$ and $M_\infty = 0.8$, the same



(a) Mean drag coefficient, μ_{C_d}



(b) Standard deviation, σ_{C_d}

Fig. 15 The optimization history of the mean and the standard deviation of the drag coefficient for the pure aleatory uncertainty case starting from two initial airfoil shapes (NACA 2412 and NACA 0012)

as in the pure aleatory case. The k factor, which is multiplied by the turbulent eddy-viscosity coefficient of the Spalart–Allmaras turbulence model, is the epistemic uncertain variable defined with the interval $[0.5, 2.0]$. The objective of the robust optimization under mixed uncertainties (Eq. (27)) is to reduce the average of the mean ($\bar{\mu}_{C_d}$), the average of the standard deviation ($\bar{\sigma}_{C_d}$), and the difference in the standard deviation of the drag coefficient ($\delta\sigma_{C_d}$) simultaneously to obtain an airfoil shape with minimum drag that is least sensitive to the change in Mach number and the k factor (i.e., the turbulence model) in the range specified for each variable. Besides the side (geometric) constraints on the design variables, the minimization is performed such that the minimum of the mean lift coefficient obtained with the optimum design is greater than or equal to 0.5. The stochastic response surfaces for the drag and the lift coefficients were again created with point-collocation

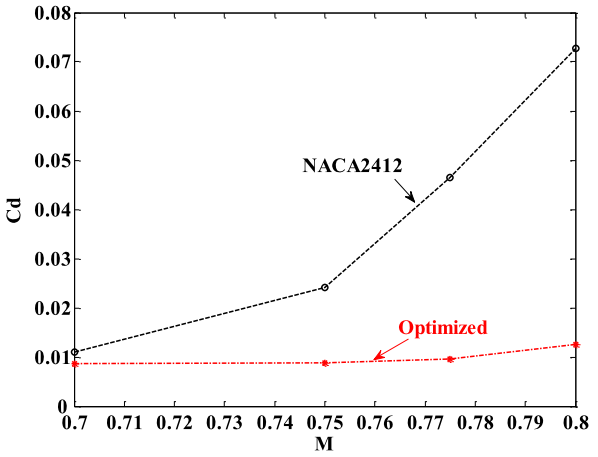


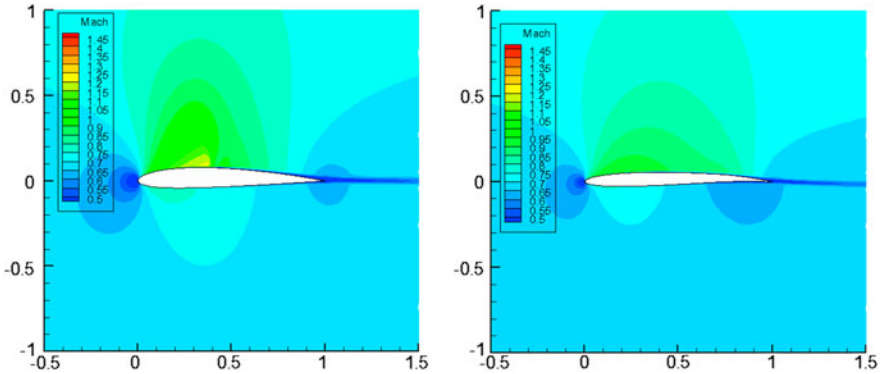
Fig. 16 The drag coefficients of the NACA 2412 and optimized airfoils at $C_L^* = 0.5$

Table 6 Drag coefficient and L/D values for NACA 2412 and optimum airfoils at various Mach numbers for pure aleatory uncertainty case

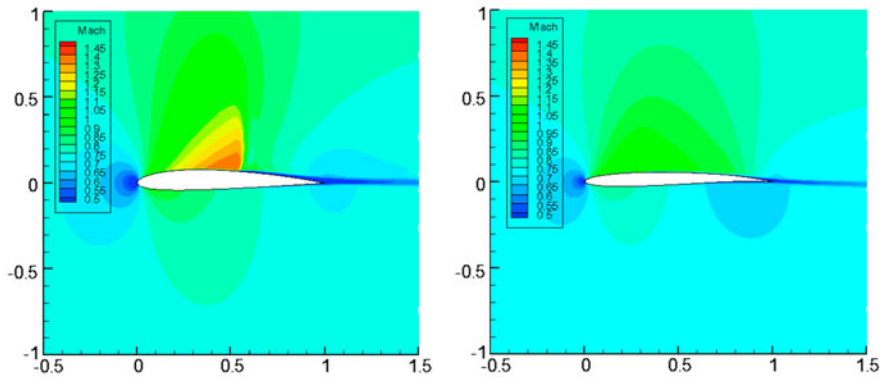
M	C_d		L/D	
	NACA 2412	Optimized	NACA 2412	Optimized
0.7	0.0110	0.0086	45.45	58.14
0.75	0.0242	0.0088	20.66	56.82
0.8	0.0727	0.0126	6.88	39.68

NIPC using a quadratic polynomial expansion with an OSR of 2 for five variables (three deterministic design variables, one aleatory uncertain variable, and one epistemic uncertain variable). The total number of CFD evaluations required for this case was $N_{\text{CFD}} = 126$, as can be calculated using Eq. (33) along with Eq. (11). Because we consider the propagation of an aleatory and epistemic uncertain variable simultaneously, this number signifies the computational efficiency of the proposed optimization approach. After the stochastic response surfaces for the drag and lift coefficients are created, the robust optimization is performed using the approach described in the previous section. As a result of the optimization under mixed uncertainties, the same optimum airfoil shape as in the pure aleatory uncertainty case is obtained (Table 7). This optimum has been verified by starting the optimization from two different initial profile shapes (NACA 2412 and NACA 0012). This result is somehow expected, since the flow field around the optimum airfoil shape does not include complex flow features such as strong shocks and shock-induced separation over the range of Mach numbers considered to make the effect of the turbulence model (i.e., the k factor) significant on different aerodynamic quantities including the drag coefficient.

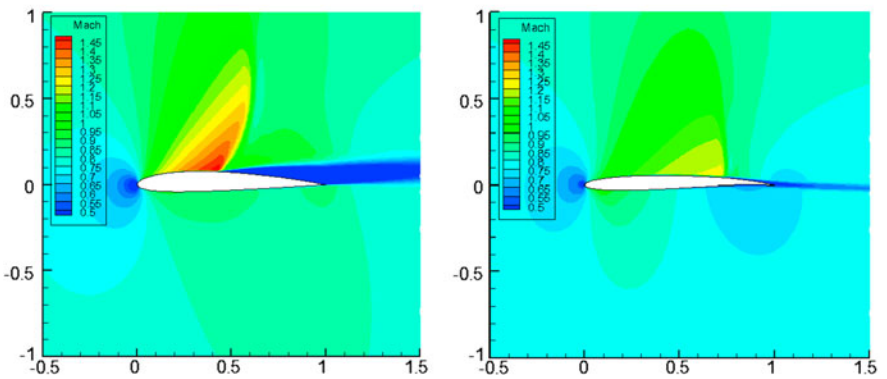
Figure 19 gives the convergence history of the average mean, average standard deviation, and the difference of the drag coefficient for the mixed uncertainty case starting from two initial airfoil shapes (NACA 2412 and NACA 0012). Regardless



(a) $M_\infty = 0.7$, NACA 2412 (b) $M_\infty = 0.7$, Optimum airfoil

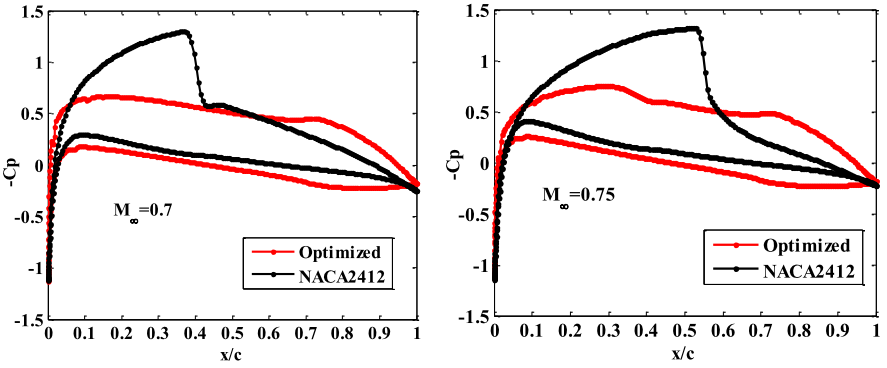


(c) $M_\infty = 0.75$, NACA 2412 (d) $M_\infty = 0.75$, Optimum airfoil

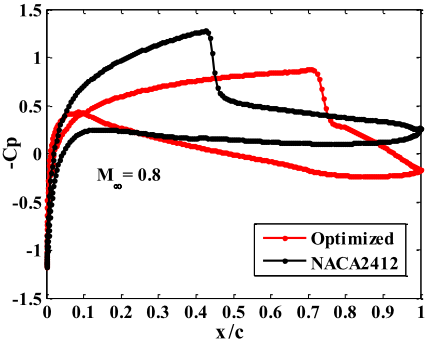


(e) $M_\infty = 0.8$, NACA 2412 (f) $M_\infty = 0.8$, Optimum airfoil

Fig. 17 Mach number contours for the NACA 2412 and optimum airfoil shapes for the pure aleatory uncertainty case



(a) $M_\infty = 0.7$ (b) $M_\infty = 0.75$



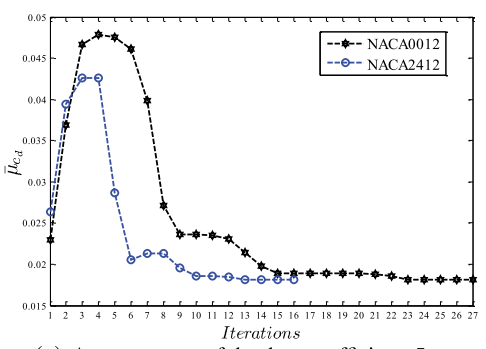
(c) $M_\infty = 0.8$

Fig. 18 The pressure distributions of the NACA 2412 and optimum airfoils at $M_\infty = 0.7$, 0.75, 0.8 for the pure aleatory uncertainty case

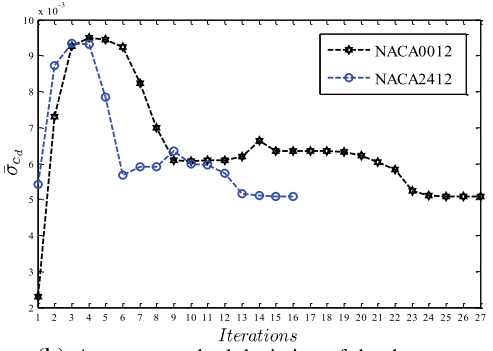
Table 7 Optimization results for the mixed uncertainty case

	Initial airfoil			Optimized airfoil		
	c	l_c	t	c	l_c	t
NACA 2412	0.020	0.400	0.120	0.019	0.700	0.080
NACA 0012	0	0	0.120	0.019	0.700	0.080

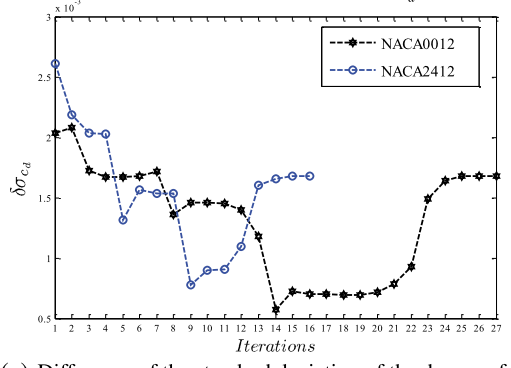
of the initial airfoil geometry used, all three quantities are reduced compared to their starting values and converge to the same final values. On the other hand, the reduction in the average mean and the average standard deviation of the drag coefficient is larger compared to the reduction in the difference of the standard deviation, which already has a small value for the initial airfoil shapes considered. This observation may imply that, for this optimization problem, the contribution of the epistemic uncertainty (i.e., k factor) is not as much as the contribution of the aleatory uncertainty (Mach number) to the total uncertainty in the drag coefficient. Since the optimiza-



(a) Average mean of the drag coefficient, $\bar{\mu}_{C_d}$



(b) Average standard deviation of the drag coefficient, $\bar{\sigma}_{C_d}$



(c) Difference of the standard deviation of the drag coefficient, $\delta\sigma_{C_d}$

Fig. 19 The optimization history of average mean, average standard deviation, and the difference of the drag coefficient for the mixed uncertainty case starting from two initial airfoil shapes (NACA 2412 and NACA 0012)

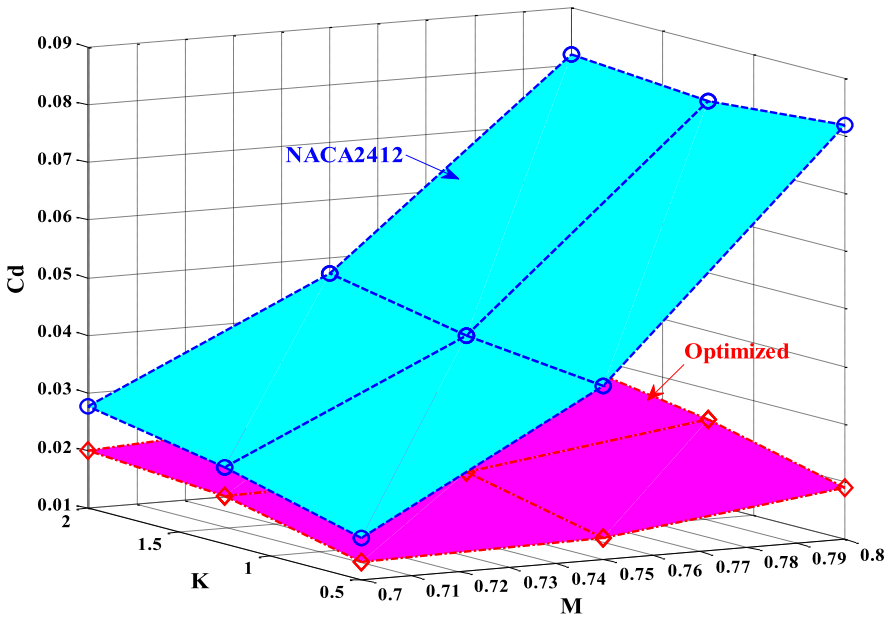


Fig. 20 Drag coefficient values of the optimized airfoil and NACA 2412 for $M_\infty = [0.7, 0.75, 0.8]$ and $k = [0.5, 1.25, 2.0]$ at $C_L^* = 0.5$

tion is performed at a relatively low lift coefficient value ($C_L^* = 0.5$), one may also expect to see more contribution from the epistemic uncertainty at higher lift coefficients. Figure 20, which shows a carpet plot of the drag coefficient over the range of M_∞ and the k factor considered, also verifies that the aerodynamic characteristics of the optimum airfoil are better compared to the characteristics of NACA 2412 (one of the airfoils used to initiate the optimization) in the case of mixed uncertainties, and no significant drag rise (i.e., variation) is observed with the optimum geometry. This plot also shows that the uncertainty in the Mach number is the main contributor to the overall uncertainty and variation in the drag coefficient, which can be quantified by the results tabulated in Table 8.

5 Conclusions

This chapter described the utilization of a computationally efficient uncertainty quantification (UQ) approach and NIPC-based stochastic expansions in robust design under mixed (aleatory and epistemic) uncertainties and demonstrated this technique on a model problem and robust aerodynamic optimization.

The optimization approach utilized stochastic response surfaces obtained with NIPC methods to approximate the objective function and the constraints in the optimization formulation. The objective function includes stochastic measures, which

Table 8 Drag coefficient and L/D values for NACA 2412 and optimum airfoils at various Mach numbers and k values for the mixed uncertainty case

K	M_∞	C_d		L/D	
		NACA 2412	Optimized	NACA 2412	Optimized
$K = 0.5$	0.7	0.0171	0.0131	29.22	38.17
	0.75	0.0401	0.0136	12.48	36.76
	0.8	0.0819	0.0189	6.11	26.48
$K = 1.25$	0.7	0.0233	0.0184	21.46	27.21
	0.75	0.0427	0.0190	11.71	26.37
	0.8	0.0798	0.0248	6.26	20.16
$K = 2.0$	0.7	0.0277	0.0201	18.07	24.88
	0.75	0.0474	0.0224	10.55	22.32
	0.8	0.0817	0.0286	6.12	17.51

are minimized simultaneously to ensure the robustness of the final design to both aleatory and epistemic uncertainties. The optimization approach was first demonstrated on the robust design of a beam under mixed uncertainties. The stochastic expansions are created with two different NIPC methods, quadrature-based and point-collocation NIPC. The optimization results were compared to the results of another robust optimization technique that utilized double-loop Monte Carlo sampling for the propagation mixed uncertainties. The results obtained with the two different optimization approaches agreed well; however, the number of function evaluations was much less than the number required by the Monte Carlo-based approach, indicating the computational efficiency of the described optimization technique.

For robust aerodynamic optimization under aleatory (Mach number) and epistemic (turbulence model) uncertainties, the NIPC response surface was also used as the basis for surrogates in the optimization process. To create the surrogates, a combined point-collocation NIPC approach was utilized, which was a function of both the design and uncertain variables. Two stochastic optimization formulations were studied: (1) optimization under pure aleatory uncertainty and (2) optimization under mixed (aleatory and epistemic) uncertainty. The formulations were demonstrated for the drag minimization of NACA four-digit airfoils described with three geometric design variables over the range of uncertainties under transonic flow conditions. Deterministic CFD simulations were performed to solve steady, two-dimensional, compressible, turbulent RANS equations. The pure aleatory uncertainty case included the Mach number as the uncertain variable. For the mixed uncertainty case, a k factor, which is multiplied by the turbulent eddy-viscosity coefficient, is introduced to the problem as the epistemic uncertain variable. The results of both optimization cases confirmed the effectiveness of the robust optimization approach with stochastic expansions by giving an optimum airfoil shape that has the minimum drag over the range of aleatory and epistemic uncertainties. The optimization under pure aleatory uncertainty case required 90 deterministic CFD evaluations,

whereas the optimization under mixed uncertainty case required 126 CFD evaluations to create the stochastic response surfaces, which show the computational efficiency of the stochastic optimization with stochastic expansions. Note also that the stochastic optimization methodology described in this chapter is general in the sense that it can be applied to any robust design problem in science and engineering.

References

1. Taguchi, G., Chowdhury, S., Taguchi, S.: Robust Engineering. McGraw-Hill, New York (2000)
2. Taguchi, G.: Taguchi on Robust Technology Development: Bringing Quality Engineering Upstream. ASME Press, New York (1993)
3. Papadimitriou, D.I., Giannakoglou, K.C.: Third-order sensitivity analysis for robust aerodynamic design using continuous adjoint. Int. J. Numer. Methods Fluids (2012)
4. Wiebenga, J.H., Van Den Boogaard, A.H., Klaseboer, G.: Sequential robust optimization of a V-bending process using numerical simulations. Struct. Multidiscip. Optim. **46**(1), 137–153 (2012)
5. Du, X., Chen, W.: Efficient uncertainty analysis methods for multidisciplinary robust design. AIAA J. **40**(3), 545–552 (2002)
6. Karpel, M., Moulin, B., Idan, M.: Robust aeroservoelastic design with structural variations and modeling uncertainties. J. Aircr. **40**(5), 946–954 (2003)
7. Ramakrishnan, B., Rao, S.S.: A general loss function based optimization procedure for robust design. Eng. Optim. **25**(4), 255–276 (1996)
8. Patel, J., Kumar, A., Allen, J.K., Ruderman, A., Choi, S.K.: Variable sensitivity-based deterministic robust design for nonlinear system. J. Mech. Des. **132**, 0645021 (2010)
9. Choi, H., McDowell, D.L., Allen, J.K., Rosen, D., Mistree, F.: An inductive design exploration method for robust multiscale materials design. J. Mech. Des. **130**(3) (2008)
10. Ruderman, A., Choi, S.-K., Patel, J., Kumar, A., Allen, J.K.: Simulation-based robust design of multiscale products. J. Mech. Des. **132**(10) (2010)
11. Ray, T., Saha, A.: Practical robust design optimization using evolutionary algorithms. J. Mech. Des. **133** (2011)
12. Li, H.X., Lu, X.: Perturbation theory based robust design under model uncertainty. J. Mech. Des. **131**, 1110061 (2009)
13. Beyer, H.-G., Sendhoff, B.: Robust optimization−a comprehensive survey. Comput. Methods Appl. Mech. Eng. **196**(33–34), 3190–3218 (2007)
14. Fishman, G.S.: Monte Carlo: Concepts, Algorithms, and Applications. Springer, New York (1995). ISBN 0-387-94527-X
15. Kroese, D.P., Taimre, T., Botev, Z.I.: Handbook of Monte Carlo Methods, p. 772. Wiley, New York (2011). ISBN 0-470-17793-4
16. Siebert, B.R.L., Cox, M.G.: The use of a Monte Carlo method for evaluating uncertainty and expanded uncertainty. Metrologia **43**(4), S178–S188 (2006)
17. Zhao, L.Y., Zhang, X.Q.: Uncertainty quantification of a flapping airfoil with a stochastic velocity deviation based on a surrogate model. Adv. Mater. Res. **201–203**, 1209–1212 (2011)
18. Zhao, L., Zhang, X.: Uncertainty quantification of a flapping airfoil with stochastic velocity deviations using the response surface method. Open Mech. Eng. J. **5**(1), 152–159 (2011)
19. Du, X.: Unified uncertainty analysis by the first order reliability method. J. Mech. Des. **130**(9), 0914011 (2008)
20. Sujecki, S.: Extended Taylor series and interpolation of physically meaningful functions. Opt. Quantum Electron., 1–14 (2012)

21. Chen, Z.-J., Xiao, H.: The Taylor series multipole boundary element method (TSM-BEM) and its applications in rolling engineering. Chongqing Daxue Xuebao **35**(5), 57–63 (2012)
22. Shu, C., Peng, Y., Zhou, C.F., Chew, Y.T.: Application of Taylor series expansion and least-squares-based lattice Boltzmann method to simulate turbulent flows. J. Turbul. **7**, 1–12 (2006)
23. Rahman, S., Rao, B.N.: A perturbation method for stochastic meshless analysis in elastostatics. Int. J. Numer. Methods Eng. **50**, 1961–1991 (2001)
24. Waiboer, R.R., Aarts, R.G.K.M., Jonker, J.B.: Application of a perturbation method for realistic dynamic simulation of industrial robots. Multibody Syst. Dyn. **13**(3), 323–338 (2005)
25. Khattri, S.K.: Series expansion of functions with He's homotopy perturbation method. Int. J. Math. Educ. Sci. Technol. **43**(5), 677–684 (2012)
26. Lee, S.H., Chen, W.: A comparative study of uncertainty propagation methods for black-box-type problems. Struct. Multidiscip. Optim. **37**(3), 239–253 (2009)
27. Eldred, M.S., Webster, C.G., Constantine, P.G.: Evaluation of non-intrusive approaches for Wiener-Askey generalized polynomial chaos. In: 10th AIAA Non-deterministic Approaches Forum, Schaumburg, IL (2008). AIAA-paper 2008-1892
28. Eldred, M.S., Burkardt, J.: Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification. In: 47th AIAA Aerospace Sciences Meeting, Orlando, FL, January (2009). AIAA 2009-0976
29. Veneziano, D., Agarwal, A., Karaca, E.: Decision making with epistemic uncertainty under safety constraints: an application to seismic design. Probab. Eng. Mech. **24**(3), 426–437 (2009)
30. Huang, H.-Z., Zhang, X.: Design optimization with discrete and continuous variables of aleatory and epistemic uncertainties. J. Mech. Des. **131**(3), 0310061 (2009)
31. Dolšek, M.: Simplified method for seismic risk assessment of buildings with consideration of aleatory and epistemic uncertainty. Struct. Infrastruct. Eng. **8**(10), 939–953 (2012)
32. Swiler, L.P., Paez, T., Mayes, R.: Epistemic uncertainty quantification tutorial. In: SAND 2008-6578C, Paper 294 in the Proceedings of the IMAC XXVII Conference and Exposition on Structural Dynamics, Society for Structural Mechanics, Orlando, FL, Feb. (2009)
33. Swiler, L., Paez, T., Mayes, R., Eldred, M.: Epistemic uncertainty in the calculation of margins. In: 50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Palm Springs, CA, May (2009). AIAA 2009-2249
34. Du, X.: Reliability-based design optimization with dependent interval variables. Int. J. Numer. Methods Eng. **91**(2), 218–228 (2012)
35. Ju, Y.P., Zhang, C.H.: Multi-point robust design optimization of wind turbine airfoil under geometric uncertainty. Proc. Inst. Mech. Eng. A, J. Power Energy **226**(2), 245–261 (2012)
36. Haro Sandoval, E., Anstett-Collin, F., Basset, M.: Sensitivity study of dynamic systems using polynomial chaos. Reliab. Eng. Syst. Saf. **104**, 15–26 (2012)
37. Didier, J., Faverjon, B., Sinou, J.-J.: Analysing the dynamic response of a rotor system under uncertain parameters by polynomial chaos expansion. J. Vib. Control **18**(5), 712–732 (2012)
38. Cheng, H., Sandu, A.: Efficient uncertainty quantification with the polynomial chaos method for stiff systems. Math. Comput. Simul. **79**(11), 3278–3295 (2009)
39. Hosder, S., Walters, R.W., Balch, M.: Efficient sampling for non-intrusive polynomial chaos applications with multiple input uncertain variables
40. Hosder, S., Walters, R.W., Perez, R.: A non-intrusive polynomial chaos method for uncertainty propagation in CFD simulations. In: 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January (2006). AIAA 2006-891
41. Bettis, B., Hosder, S.: Quantification of uncertainty in aerodynamic heating of a reentry vehicle due to uncertain wall and freestream conditions. In: 10th AIAA Joint Thermophysics and Heat Transfer Conference, Chicago, IL, June (2010). AIAA 2010-4642
42. Eldred, M.S., Swiler, L.P., Tang, G.: Mixed aleatory-epistemic uncertainty quantification with stochastic expansions and optimization-based interval estimation. Reliab. Eng. Syst. Saf. **96**(9), 1092–1113 (2011)
43. Hosder, S., Bettis, B.: Uncertainty and sensitivity analysis for reentry flows with inherent and model-form uncertainties. J. Spacecr. Rockets **49**(2), 193–206 (2012)

44. Eldred, M.S.: Design under uncertainty employing stochastic expansion methods. Int. J. Uncertain. Quantif. **1**(2), 119–146 (2011)
45. Dodson, M., Parks, G.T.: Robust aerodynamic design optimization using polynomial chaos. J. Aircr. **46**(2), 635–646 (2009)
46. Youn, B.D., Choi, K.K., Du, L., Gorsich, D.: Integration of possibility-based optimization and robust design for epistemic uncertainty. J. Mech. Des. **129**(8), 876–882 (2007)
47. Eldred, M.S.: Recent advances in non-intrusive polynomial chaos and stochastic collocation methods for uncertainty analysis and design (2009)
48. Du, X., Venigella, P.K., Liu, D.: Robust mechanism synthesis with random and interval variables. Mech. Mach. Theory **44**(7), 1321–1337 (2009)
49. Zhang, Y., Hosder, S., Leifsson, L., Koziel, S.: Robust airfoil optimization under inherent and model-form uncertainties using stochastic expansions. In: 50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, Nashville, TN, January 9–12 (2012). AIAA 2012-0056
50. Hosder, S., Walters, R.W., Balch, M.: Point-collocation nonintrusive polynomial chaos method for stochastic computational fluid dynamics. AIAA J. **48**(12), 2721–2730 (2010)
51. Wiener, N.: The homogeneous chaos. Am. J. Math. **60**(4), 897–936 (1994)
52. Xiu, D., Karniadakis, G.E.: Modeling uncertainty in flow simulations via generalized polynomial chaos. J. Comput. Phys. **187**(1), 137–167 (2003)
53. Walters, R.W., Huyse, L.: Uncertainty analysis for fluid mechanics with applications. Technical report, ICASE 2002-1, NASA/CR-2002-211449, NASA Langley Research Center, Hampton, VA (2002)
54. Najm, H.N.: Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. Annu. Rev. Fluid Mech. **41**, 35–52 (2009)
55. Hosder, S., Walters, R.W.: Non-intrusive polynomial chaos methods for uncertainty quantification in fluid dynamics. In: 48th AIAA Aerospace Sciences Meeting, Orlando, FL, January 4–7 (2010). AIAA-paper 2010-0129
56. Hosder, S., Walters, R.W., Balch, M.: Efficient sampling for non-intrusive polynomial chaos applications with multiple input uncertain variables. In: 9th AIAA Non-deterministic Approaches Conference, Honolulu, HI, April (2007). AIAA-paper 2007-1939
57. Vanderplaats, G.N.: Numerical Optimization Techniques for Engineering Design, 3rd edn. Vanderplaats Research and Development, Colorado Springs (1999)
58. Anderson, J.D.: Fundamentals of Aerodynamics, 4th edn. McGraw-Hill, New York (2010)
59. Spalart, P.R., Allmaras, S.R.: A one equation turbulence model for aerodynamic flows. In: 38th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 6–9 (1992). AIAA-paper-92-0439
60. FLUENT, ver. 13.0. ANSYS Inc., Southpointe, 275 Technology Drive, Canonsburg, PA, 15317 (2011)
61. Abbott, I.H., Von Doenhoff, A.E.: Theory of Wing Sections. Dover Publications, Mineola (1959)
62. ICEM CFD, ver. 13.0. ANSYS Inc., Southpointe, 275 Technology Drive, Canonsburg, PA 15317 (2011)

Surrogate Models for Aerodynamic Shape Optimisation

Selvakumar Ulaganathan and Nikolaos Asproulis

Abstract The main challenges in full-scale aerospace systems development are related to the level of our understanding with respect to the systems behaviour. Computational modelling, through high-fidelity simulations, provides a viable approach towards efficient implementation of the design specifications and enhancing our understanding of the system's response. Although high-fidelity modelling provides valuable information the associated computational cost restricts its applicability to full-scaled systems. This chapter presents a Computational Fluid Dynamics optimisation strategy based on surrogate modelling for obtaining high-fidelity predictions of aerodynamic forces and aerodynamic efficiency. An Aerodynamic Shape Optimisation problem is formulated and solved using Genetic Algorithm with surrogate models in the place of actual computational fluid dynamics algorithms. Ordinary Kriging approach and Hammersley Sequence Sampling plan are used to construct the surrogate models.

Keywords Surrogate models · Surrogate-based optimisation · Aerodynamic shape optimisation · Supervised machine learning

1 Introduction

The computational cost involved in performing numerical simulations for designing and optimising various engineering systems, such as aircraft wings, has been continuously increasing. From an engineering perspective, the design process is crucial for achieving maximum efficiency with the minimum possible cost and within the manufacturing restrictions. In the context of aircraft wings, aerodynamic shape optimisation (ASO) techniques are of great importance for designing a lifting surface

S. Ulaganathan (✉)
iMINDS, INTEC, Ghent University, Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium
e-mail: ulag.selva@gmail.com

N. Asproulis
Cranfield University, Cranfield MK43 0AL, UK
e-mail: n.asproulis@cranfield.ac.uk

S. Koziel, L. Leifsson (eds.), *Surrogate-Based Modeling and Optimization*,
DOI 10.1007/978-1-4614-7551-4_12,

with maximum aerodynamic efficiency. In ASO, aerodynamic constraints such as flow properties, Mach number (M), etc. alone are taken into account, in contrast to multidisciplinary design optimisation (MDO) where constraints from various (financial, structural, manufacturing, etc.) disciplines are considered. As the number of influencing disciplines increases, the complexity of the optimisation problem increases.

The constraints of the optimisation problem define the spectrum of the design variables and strongly influence the search space where the optimal solution lies. Parameterisation parameters, which will parameterise the geometry of the engineering system to be optimised, serve as a part of the design variables of the optimisation problem. Since the number of design variables directly influences the complexity of the problem, various parameterisation approaches such as the discrete point approach, partial differential equation approach and polynomial approach have been developed with the intention of simplifying the parameter space without compromising the accurate description of the geometry [1].

Once the geometry is generated, the flow-governing equations can be solved on a suitable mesh using a high- (computationally very intensive) or low-fidelity (computationally less intensive) solver based on the availability of computational resources and time. The solution algorithm is the most time-consuming part of an optimisation approach, since the entire simulation process must be repeated several times within the optimiser until an optimum solution is obtained or the entire search space is explored.

The optimisation schemes can be broadly classified into two different categories: gradient-based optimisation schemes and evolutionary optimisation schemes. The choice of the starting point becomes increasingly important in gradient-based optimisation schemes, as they are more likely to be converged or stuck into local optima [2, 3]. The non-derivative methods are more powerful in finding the global optimum within the given search space; however they are lacking in terms of finding the exact global optimum [2, 3]. These characteristics are given by the concepts of *exploration* (ability to find the location of the global optimum without getting trapped in a local extremum) and *exploitation* (ability to exploit the exact optimum solution).

The cost of the ASO process increases by several orders of magnitude when the actual, computationally expensive solution algorithms are employed for resolving the fluid flow [4]. Consequently, the computational cost becomes prohibitively expensive, and a need arises to use well-tuned optimisation methods in order to efficiently identify the optimum configurations within the design space. The computational limitations become more apparent when numerically demanding solution methods, such as computational fluid dynamics (CFD), are coupled with the optimiser. As a consequence, the fidelity of the analysis at the early stages of the design process is sacrificed with the aim of reducing the overall computational burden. In order to circumvent this problem, the use of approximation models has become more popular in recent years in imitating complex solution algorithms due to their quick response and reduced computing requirements. Within this framework, approximation models are constructed for computationally demanding solution algo-

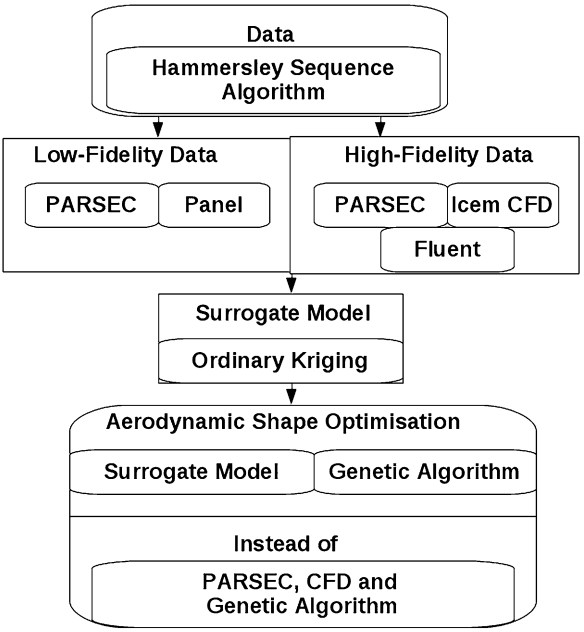


Fig. 1 Work flow of the problem

rithms and, further, are used in the place of actual solution algorithms during the optimisation [5, 6].

The process of constructing an approximation model usually involves the following steps: (a) generation of computational data, (b) learning from the collected computational data, and (c) constructing a surrogate model based on the learning. Various approaches such as polynomial regression, response surfaces, neural networks and kriging are used to construct the approximation models [4, 7].

In this chapter, in order to show the advantage of surrogate-assisted optimisation, an ASO problem is formulated to identify the best possible airfoil geometry which will have an improved aerodynamic efficiency for the given flow, structural and aerodynamic conditions. The aerodynamic efficiency (E) is defined as the ratio of lift to drag. Lift and drag are the vertical and horizontal forces, respectively, which act on an airfoil when it is introduced into the airflow. These forces are primarily responsible for the aerodynamic efficiency of an airfoil. The example problem formulated in this chapter considers the NACA 2411 airfoil geometry as the baseline shape to be optimised. The airfoil is assumed to be introduced into a viscous, compressible and low turbulence airflow with M varying between 0.1 and 0.6 at a fixed angle of attack of 5.0°. The formulated problem is solved to optimise the baseline airfoil in the assumed airflow conditions. Figure 1 depicts the work flow involved in solving the formulated optimisation problem.

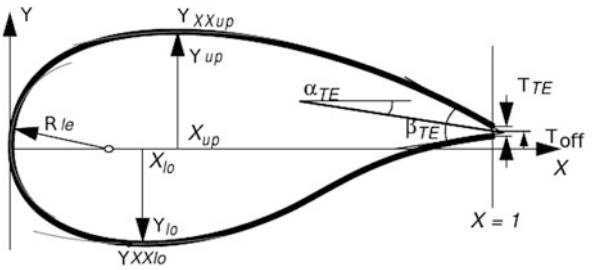


Fig. 2 PARSEC control parameters

2 Methodology

This section provides an overview of the methodology used in this chapter and is structured as follows. Section 2.1 describes the parameterisation method employed, called PARSEC, Sect. 2.2 describes the sampling algorithm used, Sect. 2.3 describes the construction of approximation models using the OK approach, and finally Sect. 2.4 provides the overall optimisation procedure.

2.1 Parameterisation

PARSEC is a parameterisation scheme which describes the lower and the upper surface of an airfoil independently using a sixth order polynomial [8]. In this approach, the shape of the airfoil is controlled by the following 11 parameters [9, 10]: leading edge radius (R_{le}), upper crest point (y_{up}), lower crest point (y_{lo}), position of upper crest (x_{up}), position of lower crest (x_{lo}), upper crest curvature ($y_{xx_{\text{up}}}$), lower crest curvature ($y_{xx_{\text{lo}}}$), trailing edge thickness (T_{te}), trailing edge offset (T_{off}), trailing edge wedge angle (β_{te}), trailing edge direction angle (α_{te}). These parameters are shown in Fig. 2 [8].

R_{le} is divided into lower leading edge radius (R_{leu}) and upper leading edge radius (R_{lel}) in order to increase the accuracy of the method near the leading edge. Hence, 12 design parameters are used instead of the typical 11 parameters [11].

The mathematical formulation of the approach is given by Eqs. (1) and (2) for the upper and lower surfaces of the airfoil, respectively [1, 11, 12].

$$y_u = \sum_{i=1}^{6} a_i x^{i-(1/2)}, \tag{1}$$

$$y_l = \sum_{i=1}^{6} b_i x^{i-(1/2)}, \tag{2}$$

where, y_u is the required y co-ordinate for the upper surface, y_l is the required y co-ordinate for the lower surface, x is the non-dimensional chord-wise location

(chord (c) is assumed to be 1) and a_i and b_i are the coefficients to be solved. The surface of the airfoil is obtained from the solution of the above two equations subject to the following geometrical conditions: (1) at $x = \text{maximum}$, $y = \text{maximum}$, (2) at $x = \text{maximum}$, $\frac{dy}{dx} = 0$, (3) at $x = \text{maximum}$, $\frac{d^2y}{dx^2} = \text{maximum}$, (4) at $x_{\text{up}} = 1$, $y_{\text{up}} = T_{\text{off}} + \frac{T_{\text{TE}}}{2}$, (5) at $x_{\text{lo}} = 1$, $y_{\text{lo}} = T_{\text{off}} - \frac{T_{\text{TE}}}{2}$, (6) at $x_{\text{up}} = 1$, $\frac{dy_{\text{up}}}{dx} = \tan(\alpha_{\text{TE}} - \frac{\beta_{\text{TE}}}{2})$, (7) at $x_{\text{lo}} = 1$, $\frac{dy_{\text{lo}}}{dx} = \tan(\alpha_{\text{TE}} + \frac{\beta_{\text{TE}}}{2})$.

2.2 Sample Generation

The Hammersley sequence sampling (HSS) technique is a low-discrepancy sampling approach that generates N sample points in a k-dimensional hypercube [13]. Each sample point that falls within the design space constitutes a design point by defining the design variables. For measuring the deviation of the generated sample points from a uniform distribution a quantitative criterion is employed, called the discrepancy [14]. It is always desired to have a more uniform distribution of the sample points within the design space, since it increases the efficiency of the learning from the collected data during the construction of surrogate models. An extensive description regarding the HSS technique can be found in Kalagnanam and Diwekar [15].

In this approach, an integer n is represented by the radix-R notation as shown below [15]:

$$n \equiv n_m n_{m-1} \cdots n_2 n_1 n_0, \tag{3}$$

$$n = n_0 + n_1 R + n_2 R^2 + \cdots + n_m R^m, \tag{4}$$

where $m = [\log_R n] = [\frac{\ln n}{\ln R}]$ is the integer portion of $\log_R n$. For example, the integer 1,256 has $p_0 = 6$, $p_1 = 5$, $p_2 = 2$, $p_3 = 1$, $R = 10$ and $m = 3$ in the radix-10 notation [14]. The inverse radix number, which is defined as a unique fractional value located between 0 and 1, is obtained by reversing the order of the digits of the integer around the decimal point [14]:

$$\phi_R(n) = .n_0 n_1 n_2 \cdots n_m, \tag{5}$$

$$\phi_R(n) = n_0 R^{-1} + n_1 R^{-2} + \cdots + n_m R^{-m-1}. \tag{6}$$

The HSS algorithm generates N sample points in a k-dimensional hypercube using the following relation [14]:

$$x_k(n) = \left(\frac{n}{N}, \phi R_1(n), \phi R_2(n), \ldots, \phi_{R_{k-1}}(n) \right),$$
$$n = 0, 1, 2, \ldots, N-1, \tag{7}$$

where $R_1, R_2, \ldots, R_{k-1}$ represent the first $k-1$ prime numbers. For the opted problem, ten PARSEC parameters (T_{te} and T_{off} are fixed due to structural and aerodynamic constraints) together with M serve as the design variables.

2.3 Surrogate Model Construction

Kriging techniques are employed for interpolations of random responses and are based on stochastic processes. In the case of ordinary kriging (OK), the mathematical expression for the function is defined as

$$\hat{f}(x_p) = \sum_{i=1}^{N} \gamma_i(x_p) f(x_i) \quad \forall x_p \in S, \tag{8}$$

where $\hat{f}$ is the linear estimator function for f, $\gamma_i(x_p)$ is the weighting function and x_p is a vector of sample points in the design space, which in our case is defined through the range of values of ten PARSEC parameters along with M and is denoted by $S \subset \mathbb{R}^{11}$.

Since the OK model is an isotropic stationary model [16], it is implied that the covariance of f between two sample points is described by a function which is solely based on the distance between the two sample points rather than their exact locations. It can be expressed as

$$C\big[f(x_a), f(x_b)\big] = C\big(|x_a - x_b|\big). \tag{9}$$

$C[f(x_a), f(x_b)]$ is often expressed by the covariance matrix as [16]

$$\underline{C} = \begin{Bmatrix} \sigma^2 = C(0) & C(\|x_1 - x_2\|) & \dots & C(\|x_1 - x_N\|) \\ C(\|x_2 - x_1\|) & \sigma^2 & \dots & C(\|x_2 - x_N\|) \\ \vdots & \vdots & \ddots & \vdots \\ C(\|x_N - x_1\|) & C(\|x_N - x_2\|) & \dots & \sigma^2 \end{Bmatrix}, \tag{10}$$

where σ^2 is the variance of the sample points. For the unknown sample point $x_p \in S$, the covariance vector ($\overrightarrow{c}$) and the weighting functions vector can be expressed as [16]

$$\overrightarrow{c}(x_p) = \begin{Bmatrix} C(\|x_p - x_1\|) \\ C(\|x_p - x_2\|) \\ \vdots \\ C(\|x_p - x_N\|) \end{Bmatrix}; \qquad \gamma_i(x_p) = \begin{Bmatrix} \gamma_1(x_p) \\ \gamma_2(x_p) \\ \vdots \\ \gamma_N(x_p) \end{Bmatrix}. \tag{11}$$

Since it is an isotropic stationary model, the sum of all the weighting functions should be equal to 1, as given in Eq. (12). Hence the covariance matrix and covariance vector are then expressed as Eqs. (13) and (14), respectively.

$$\sum_{i=1}^{N} \gamma_i(x_p) = 1 \quad \forall x_p \in S, \tag{12}$$

$$\underline{C} = \begin{Bmatrix} \sigma^2 = C(0) & C(\|x_1 - x_2\|) & \dots & C(\|x_1 - x_N\|) & 1 \\ C(\|x_2 - x_1\|) & \sigma^2 & \dots & C(\|x_2 - x_N\|) & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ C(\|x_N - x_1\|) & C(\|x_N - x_2\|) & \dots & \sigma^2 & 1 \\ 1 & 1 & \dots & 1 & 0 \end{Bmatrix}, \tag{13}$$

$$\mathbf{c}(x_p) = \begin{Bmatrix} C(\|x_p - x_1\|) \\ C(\|x_p - x_2\|) \\ \vdots \\ C(\|x_p - x_n\|) \\ 1 \end{Bmatrix}. \tag{14}$$

A Lagrange multiplier (λ_{x_p}) is introduced in the weighting functions vector in order to enforce the unbiasedness constraint of the OK model. Hence the weighting functions vector becomes:

$$\gamma_i(x_p) = \begin{Bmatrix} \gamma_1(x_p) \\ \gamma_2(x_p) \\ \vdots \\ \gamma_N(x_p) \\ \lambda_{x_p} \end{Bmatrix}. \tag{15}$$

The weighting functions are calculated using the covariance matrix and covariance vector as given by the following relation:

$$\gamma_i(x_p) = \underline{C}^{-1}\mathbf{c}(x_p). \tag{16}$$

Since the predicted value of the response at an unknown sample point is always different from the actual value at that sample point, an error measure is introduced to measure the prediction capability of the OK model. This measure of error is known as the estimation error (e_p) and is defined as follows:

$$e_p = \hat{f}(x_p) - f(x_p), \tag{17}$$

where $f(x_p)$ is the actual value at an unknown point $x_p \in S$. If the weighting functions are obtained in such a way that they will reduce the variance of the estimation error, then a function predictor with optimal prediction capability can be obtained. The error variance can be computed using the following expression:

$$V(e_p) = \sum_{i=1}^{N} \big(\mathbf{c}(x_p)\gamma_i(x_p)\big). \tag{18}$$

Since the covariogram function is arbitrarily computed from the observed data, a suitable theoretical variogram model should be used to fit the experimental vari-

ogram model, so that the kriging equations become solvable. Generally, the selection of a suitable theoretical variogram model is carried out using maximum likelihood estimation (MLE) or cross-validation (CV) approaches. For the current problem, the following theoretical variogram models are employed and the most suitable one is selected based on the MLE approach.

- Gaussian model with actual range:

$$C(h) = sill\left(1 - \exp\left(\frac{-h^2}{range^2}\right)\right) \tag{19}$$

- Gaussian model with practical range:

$$C(h) = sill\left(1 - \exp\left(\frac{-3h^2}{prange^2}\right)\right) \tag{20}$$

- Spherical model with actual range:

$$C(h) = sill\left(\left(\frac{3.0}{2.0}\right)\left(\frac{h}{range}\right) - 0.5\left(\frac{h}{range}\right)^3\right) \tag{21}$$

- Exponential model with actual range:

$$C(h) = sill\left(1 - \exp\left(\frac{-h}{range}\right)\right) \tag{22}$$

- Exponential model with practical range:

$$C(h) = sill\left(1 - \exp\left(\frac{-3h}{prange}\right)\right), \tag{23}$$

where h is the isotropic lag defined as the distance between two sample points in S. In the semivariogram, the lag value at which the semivariance becomes constant is called the range, and the corresponding semivariance value is called the sill. The practical range is the value of the lag at which 0.94 % of the sill is achieved.

2.4 Aerodynamic Shape Optimisation

A genetic algorithm (GA), which is one of a class of evolutionary algorithms where the evolution is based on the theory of the *mechanics of natural selection and the evolution process*, is used to carry out the optimisation problem. Here the optimisation parameters are described by a group of genes called chromosomes [17–20]; each chromosome is a binary string which describes an individual (i.e. a sample point). In the current problem, the PARSEC parameters T_{te} and T_{off} are fixed during the optimisation along with the flow parameter α. This is done to avoid the evolution

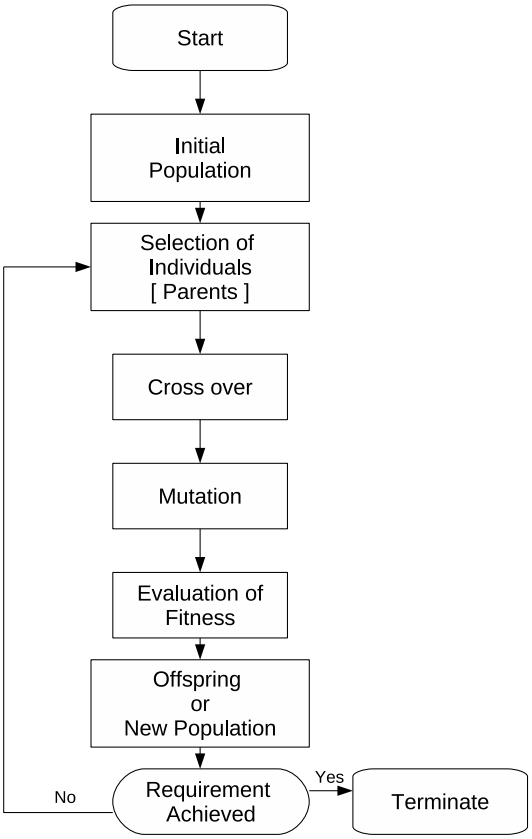


Fig. 3 Genetic algorithm approach

of airfoils with trailing edge thickness and trailing edge offset during the optimisation. The parameter α is fixed because the optimisation procedure is carried out for a fixed value of ($\alpha = 5.0°$) angle of attack. These two fixed parameters will also ensure the satisfaction of the structural and aerodynamic constraints of the current problem. Hence, the remaining ten PARSEC parameters along with the flow parameter, M, serve as the optimisation parameters in the current problem. The typical work flow of the GA is depicted in Fig. 3 and is discussed further below.

2.4.1 Search Space

The search space for the current problem is defined by the range of values of the ten PARSEC parameters and M and their required decimal accuracy. For each optimisation parameter, the required accuracy of the decimal place (d) can be specified.

Once d is specified, the domain length for a particular optimisation parameter can be expressed as follows [21]:

$$Domain\ length = 10^d (X_u - X_l), \tag{24}$$

where X_u and X_l are the upper and lower bound values of the optimisation parameter, respectively. The binary string corresponding to the parameter X is expressed as $b_{m-1}, b_{m-2}, \ldots, b_1, b_0$, which will be equal to $X' = \sum_{i=1}^{m} (b_i 2^i)$. The value of m, which is defined to specify the number of possibilities for a given 'd', can be chosen as [21]

$$2^{m-1} < 10^d (X_u - X_l) < 2^m. \tag{25}$$

All the optimisation parameters are represented as binary strings of length (L_i) and combined as a single binary string. The length of the single binary string (L_k) can be calculated as follows:

$$L_k = \sum_{i=1}^{k} L_i, \tag{26}$$

where k is the number of optimisation parameters, which is equal to n in the current problem. The first L_1 binary strings of the single binary string (L_k) correspond to the first optimisation parameter, the second L_2 binary strings correspond to the second optimisation parameter and so on.

2.4.2 Initial Population

The random number approach is employed to generate the initial pool of optimisation parameters. In this approach, a random number is generated between 0 and 1. If the random number is between 0 and 0.5, then the bit is considered as 0, whereas if it lies between 0.5 and 1, then the bit is set to 1. The size of the initial population can be controlled by a parameter called *popsize*.

2.4.3 Selection of Parents

Individuals (containing the optimisation parameters) are selected from the pool of the initial population and placed into the mating pool. These individuals are further used for mating and generating new offspring. Since the characters of these individuals are passed to the next generation, only the individuals who have desirable properties are selected. This is accomplished by the tournament wheel selection technique. In this approach, a tournament is defined among the individuals by specifying a selection pressure. The individuals with higher fitness are considered as winners of the tournament and will be placed in the mating pool [22]. The fitness function $(P(i))$ is evaluated by calculating the total objective function (F) as

follows:

$$f(i) = obj(V_i), \quad i = 1, popsize, \tag{27}$$

$$F = \sum_{i=1}^{popsize} \big(f(i)\big); \qquad P(i) = \frac{f(i)}{F}, \tag{28}$$

where $f(i)$ (different from the $f(x)$ defined in the OK section) is the objective function to be optimised and V is the set of optimisation parameters. The process of selection holds some important properties: (a) Best individuals are preferred but not always selected. (b) Worst individuals are not always excluded in order to maintain the variability in each generation.

2.4.4 Crossover

Crossover is performed to combine the desirable characters of two different parents who are selected for mating. The method of crossover depends on the kind of problem to be solved and the method of encoding. For the current problem, the uniform crossover approach is employed. In this approach, a *crossover probability* (p_c) is defined and a probability test is performed for each bit in the bit string. If passed, then the bits are randomly exchanged between the two parents selected for mating [23].

2.4.5 Mutation

Mutation is performed in order to refine the process of mating. Here a *mutation probability* (p_m) is defined and the probability test is performed on each bit in the bit stream. If passed, the bit is flipped directly. If not passed, the bit is generated randomly and compared with the current one. If the randomly generated bit is different from the original bit, then the original bit is flipped [23].

2.4.6 Fitness Evaluation

Fitness evaluation is the process of evaluating the objective function for each set of optimisation parameters. Based on the fitness of the new offspring, they are considered as new parents and selected for further mating. This process is repeated until the convergence is achieved. The following fitness criterion is used in the current problem to select the best possible airfoil geometries:

$$Fitness(f) = obj(E) = obj\left(\frac{(C_l)_{\text{High-fidelity}}}{(C_d)_{\text{High-fidelity}}}\right), \tag{29}$$

where $(C_l)_{\text{High-fidelity}}$ and $(C_d)_{\text{High-fidelity}}$ are the high-fidelity coefficient of lift and the high-fidelity coefficient of drag, respectively. The method of estimating their values is described in the following section.

Table 1 Design variables and their ranges of values

Var. No.	Design variable	Lower bound	Upper bound
01	R_{leu}	0.020	0.023
02	R_{lel}	0.006	0.010
03	x_{up}	0.300	0.380
04	y_{up}	0.077	0.080
05	yxx_{up}	−0.630	−0.650
06	x_{lo}	0.160	0.180
07	y_{lo}	−0.032	−0.034
08	yxx_{lo}	0.660	0.680
09	α_{te}	−4.700	−4.800
10	β_{te}	15.00	15.10
11	M	0.100	0.600

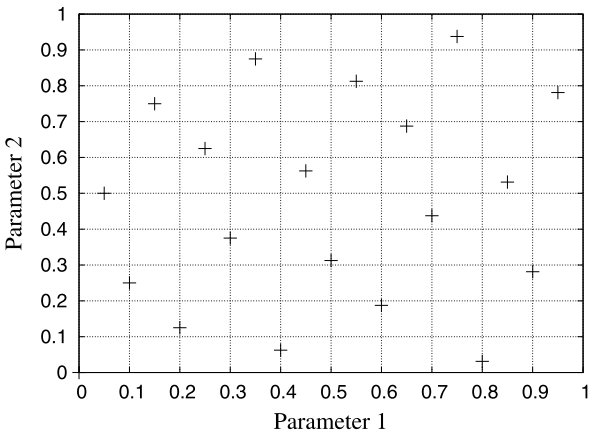


Fig. 4 Number of HS sample points: 20

3 Results and Analysis

Computer-based simulations must be performed at the optimal sample points generated by the HSS algorithm to obtain data. The collected data are used to initiate the learning process during the construction of surrogate models. As discussed earlier, the dimension (n) and the design space of the current problem are 11 and $S \subset \mathbb{R}^{11}$, respectively. Table 1 gives the design variables and their ranges of values for the current problem.

A sample point has ten PARSEC parameters and M. 50 (N) such sample points are generated where the simulations need to be performed. The HSS technique generates uniform sample points in an unstructured way. Figures 4 and 5 show the uniformity and space-filling properties of the Hammersley sequence (HS) sample points. These sample points are generated for a two-dimensional problem defined

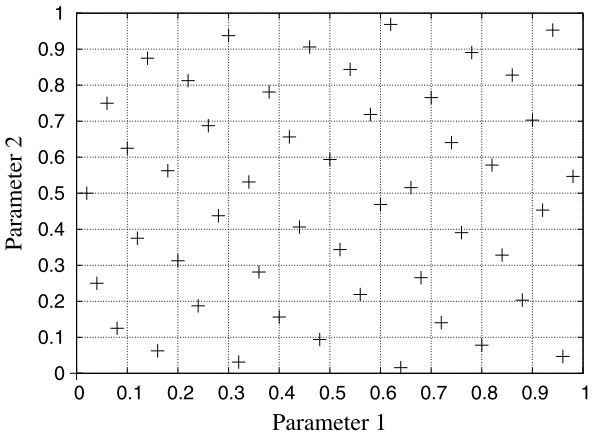


Fig. 5 Number of HS sample points: 50

in the design space of $[0, 1]^2$. It can be seen from these figures that the HSS technique retains its uniformity and space-filling properties irrespective of the number of sample points. Observe also that the sample points are spread over the interior of the design space for the given number of "N" sample points in contrast to the classical design of experiments (DOE) techniques, where the sample points are generated mainly near the boundaries of the design space [14].

Computer-based simulations, both panel (low-fidelity) and CFD (high-fidelity), are performed at these 50 sample points. A Linear Vorticity Surface Panel method code developed by Ilan Kroo [24] is used for the low-fidelity simulations. Panel methods are more effective in giving reasonably accurate results without being computationally expensive. The flow around the NACA 2411 airfoil is solved using the panel code for different angles of attack with $N_p = 1{,}000$ (number of panels). The results (C_l) (which are theoretically valid at $M = 0.0$) are compared with the Xfoil viscous simulation results obtained at $M = 0.1$, as shown in Fig. 6. The influence of N_p on the low-fidelity results is shown in Fig. 7. It can be seen that the panel method slightly over-predicts the C_l, and the accuracy of the solution increases as the number of panels increases.

High-fidelity, CFD simulations are performed by solving two-dimensional, steady and compressible Navier–Stokes equations using the FLUENT software [25]. The turbulence phenomena have been modelled using the Spalart–Allmaras turbulence model, which is a one-equation model for solving the turbulent viscosity transport equation [26, 27] and has been widely used for aerospace applications. The computational grid is generated with the ICEM CFD package. The C-grid topology is used since it is quite good at capturing the flow physics in the wake region of the airfoil [28, 29]. Figure 8 shows the topology of the grid and the dimensions involved with the grid generation. The grid extends to a dimension of $14c$ in the downstream direction (L), $9c$ in the upstream direction (A) and $10c$ in the cross-stream direction (H). In order to capture the flow physics within the boundary layer region, $y^+ = 1$ has been used [30].

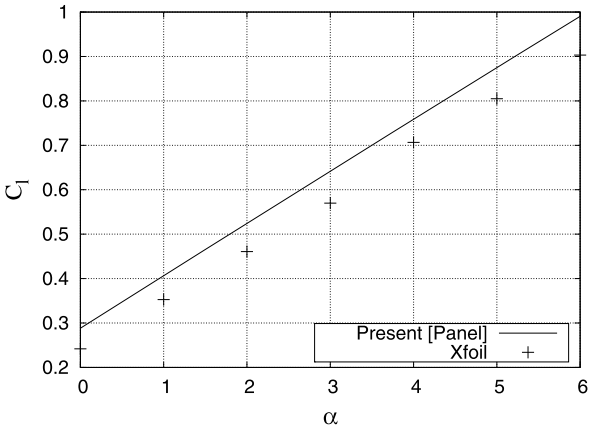


Fig. 6 C_l as a function of α

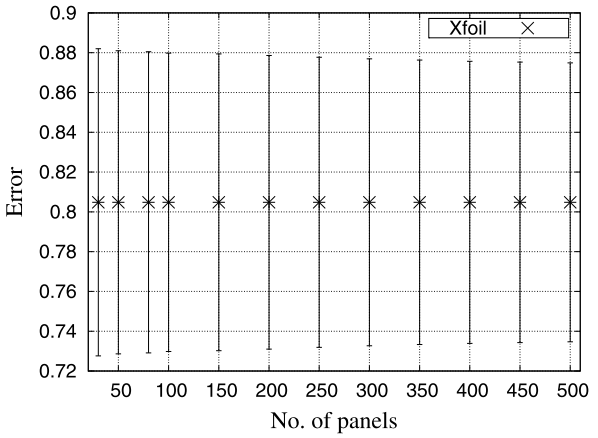


Fig. 7 Influence of number of panels on C_l

The density-based implicit solver in FLUENT is used to solve the flow around the airfoil geometry with an ideal gas as a working fluid. The viscosity is calculated from the three-coefficient Sutherland law, and the basic flow properties are given in Table 2. Turbulence is specified in terms of turbulent intensity (I) and turbulent length scale (l), and a least squares cell-based discretisation scheme is used for the gradient together with the Roe-FDS flux type. Third order Monotone Upstream-centred Schemes for Conservation Laws (MUSCL), which can provide more accurate numerical results even when the solutions exhibit shock, are employed for the spatial discretisation of the flow [31, 32]. The solution converges down to an accuracy of 10^{-5} and 10^{-6} in about 3,000 iterations.

In order to validate the mesh generation and solution techniques, the flow over the NACA 2411 and NACA 0012 airfoils is solved using the above-described mesh

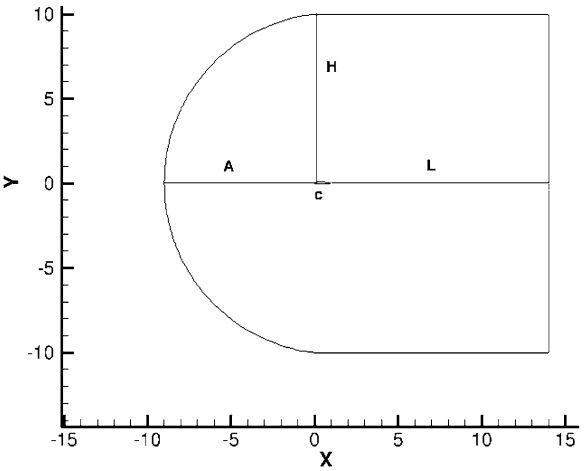


Fig. 8 Grid topology

Table 2 Fluid flow properties

Property	Value
Pressure (p)	101,325 N/m^2
Density (ρ)	1.1766 Kg/m^3
Temperature (T)	300 K
AOA (α)	5.0 deg
Mach (M)	From sample point

generation and solution methods. The flow properties for the current validation case are the same as tabulated above except for α and M. The validation is carried out for various α at $M = 0.1$. The fine grid, which is obtained after a grid convergence study, has 1,000 points on the surface of the airfoil in the circumferential direction and has about 80,000 cells in total. A two-stage boundary layer is used in the grid generation to have more cells around the geometry of the airfoil (see Fig. 9). Figures 10 and 11 show the error (variation of CFD results from the actual results) of C_l and C_d for the NACA 2411 and NACA 0012 airfoils, respectively. One can see that the error increases when the number of cells (N_c) goes above 8,000. Figure 12 shows the variation of C_l from the results published by Klimas and Sheldahl in Ref. [33] for NACA 2411 with fine mesh ($N_c = 8{,}000$). Figure 13 shows the variation of C_d from the Xfoil viscous simulation results for NACA 0012 with fine mesh.

The grid generation process for the remaining 50 sample points is automated, so that the same grid generation technique can be applied for all the airfoil geometries.

It is also ensured that the applied grid generation technique results in a fine mesh for all the airfoil geometries. The flow around the 50 airfoil geometries is solved using both the low-fidelity panel simulations and high-fidelity CFD simulations. Once the aerodynamic forces ($(C_l)_{\text{Low-fidelity}}$, $(C_l)_{\text{High-fidelity}}$ and $(C_d)_{\text{High-fidelity}}$)

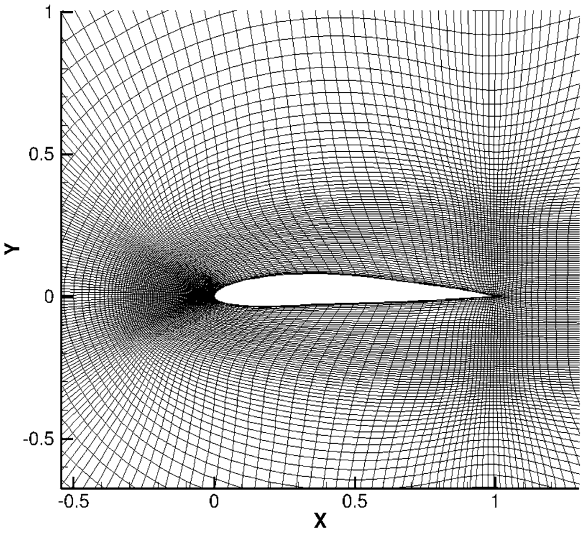


Fig. 9 Two-stage boundary layer

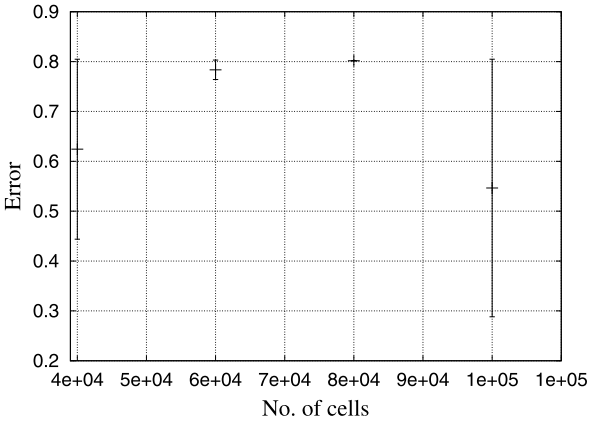


Fig. 10 Grid convergence [C_l] error estimation

are obtained for the generated 50 airfoil geometries, they can then be used for the learning process.

Three *surrogate models* are constructed using the in-house OK code. The first surrogate model is constructed using the low-fidelity panel data and can predict the low-fidelity C_l for any airfoil geometry placed within the design space S. The second surrogate model is constructed using the high-fidelity C_d data and can predict the high-fidelity C_d for any airfoil geometry placed within S. The third one is constructed using the difference in C_l between the low- and high-fidelity data (ΔC_l)

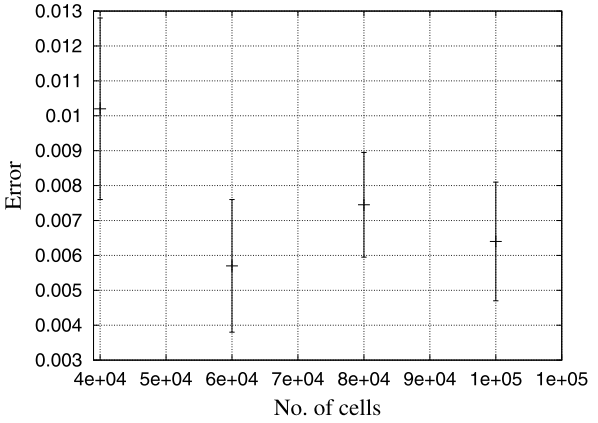


Fig. 11 Grid convergence $[C_d]$ error estimation

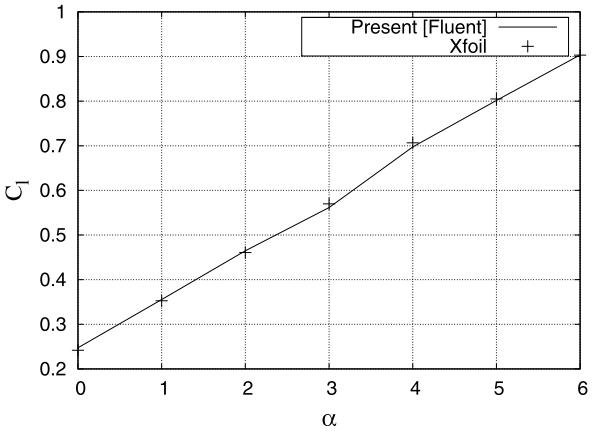


Fig. 12 C_l as a function of α for NACA 2411

and can be used to estimate the difference in C_l between the low- and high-fidelity analysis for any airfoil geometry placed within S.

Figures 14, 15 and 16 show the capability of different theoretical semivariogram models in fitting the experimental semivariogram model for the first, second and third black surrogate models respectively. It can be clearly observed that the Gaussian model with practical range fits the experimental semivariogram model more accurately than any other theoretical models for all the three surrogate models. The second most accurate one is the spherical model with actual range. These conclusions are confirmed by carrying out the prediction at various unknown sample points. Figures 17, 18 and 19 show the comparison at some of these unknown sample points for all three surrogate models with $N = 50$. For all the predictions, the e_p is on the order of 10^{-2} and 10^{-3}.

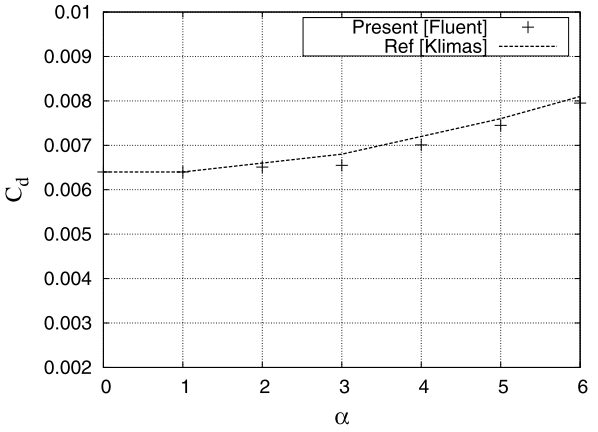


Fig. 13 C_d as a function of α for NACA 0012

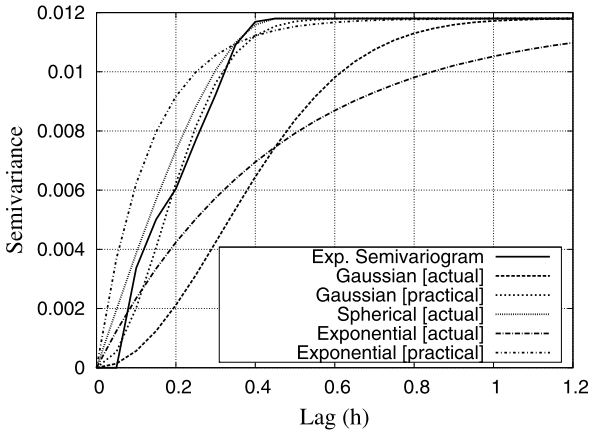


Fig. 14 Theoretical semivariogram models [low-fidelity C_l]

The number of sample points (N) (and so the amount of data) which is used to construct the surrogate models has a huge influence on the prediction capability of the constructed surrogate models. As "N" increases, the capability of the surrogate model to predict the right solution increases until a saturation level is reached for "N". This behaviour is further depicted in Figs. 20, 21 and 22 for the first, second and third surrogate models, respectively. It can be observed that e_p is reduced as "N" increases for all three surrogate models.

The aerodynamic efficiency (E) of an airfoil geometry which is placed within 'S' can be calculated using the constructed surrogate models. Once an unknown sample point (airfoil geometry and M) is generated, it can be supplied to the three surrogate models. As discussed earlier, the first surrogate model can predict the low-fidelity C_l, while the second one can predict the high-fidelity C_d. The ΔC_l can be predicted

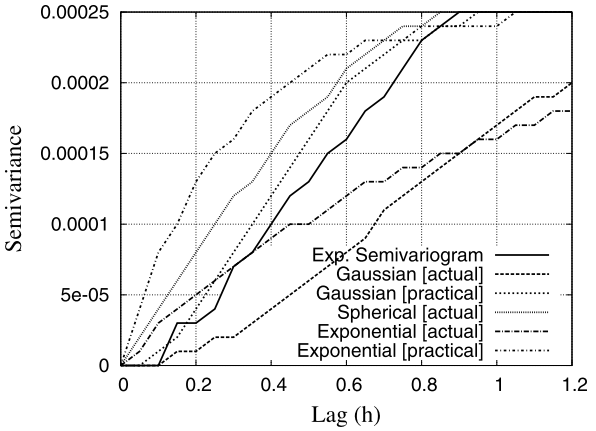


Fig. 15 Theoretical semivariogram models [high-fidelity C_d]

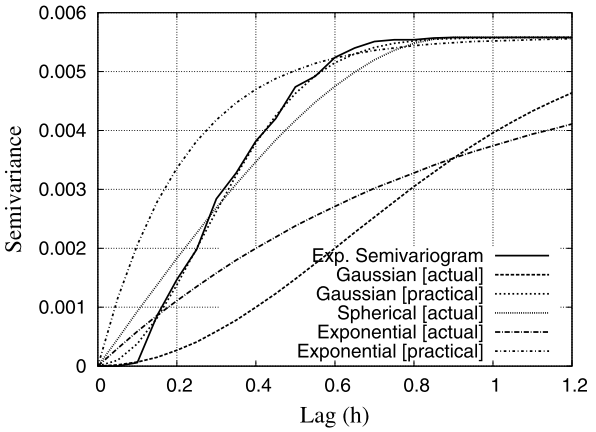


Fig. 16 Theoretical semivariogram models [ΔC_l]

by the third surrogate model. Now the E of the airfoil at $\alpha = 5.0°$ for the above discussed flow conditions can be calculated from the following relations. Since the airfoil is placed within 'S', the M will have a value between 0.1 and 0.6.

$$(C_l)_{\text{High-fidelity}} = (C_l)_{\text{Low-fidelity}} - \Delta C_l, \tag{30}$$

$$E = \frac{L}{D} = \frac{(C_l)_{\text{High-fidelity}} q_\infty S}{(C_d)_{\text{High-fidelity}} q_\infty S}, \tag{31}$$

where L is the lift force of the airfoil, D is the drag force of the airfoil, $q = (\frac{\rho V^2}{2})$ is the dynamic pressure of the flow, ρ is the density of the flow, V is the velocity of the flow and S is the surface area of the airfoil. Since S and q are constant for a given airfoil and flow conditions (M, ρ, *temperature*) respectively, the above relation can

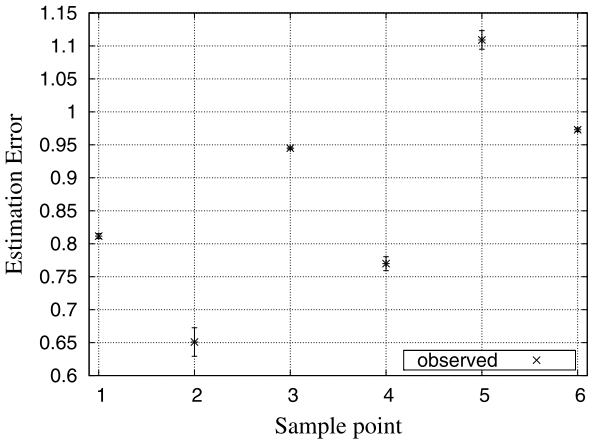


Fig. 17 Estimation error of the predictions [low-fidelity C_l]

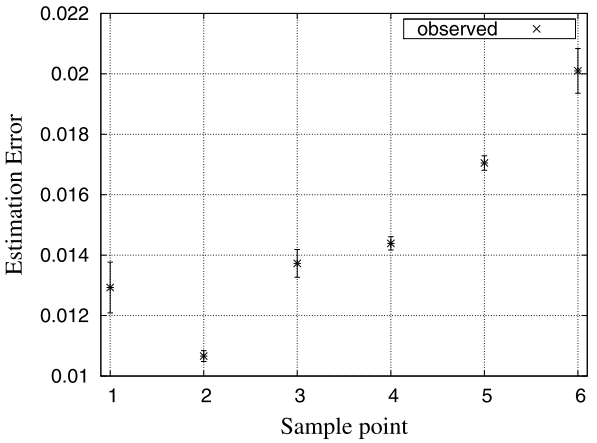


Fig. 18 Estimation error of the predictions [high-fidelity C_d]

be written as follows:

$$E = \frac{L}{D} = \frac{(C_l)_{\text{High-fidelity}}}{(C_d)_{\text{High-fidelity}}}. \tag{32}$$

E is estimated at various sample points (i.e. airfoil geometries) placed within the design space 'S' using the constructed surrogate models. The estimated values are compared with the actual values of E which are calculated from separate CFD simulations. The comparison is shown in Fig. 23.

Figure 23 shows that the prediction of the constructed surrogate models leads to having E within low error bounds with the maximum % of error being smaller

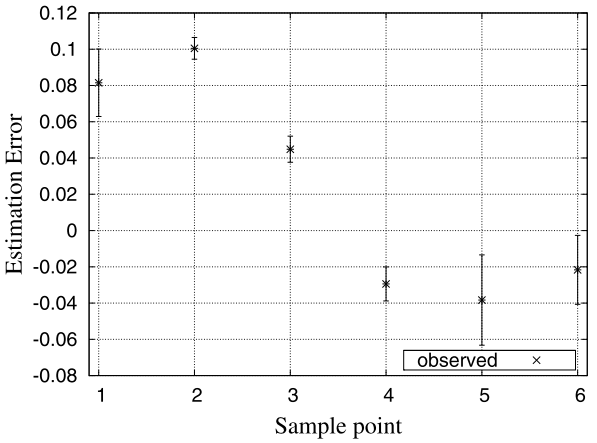


Fig. 19 Estimation error of the predictions [ΔC_l]

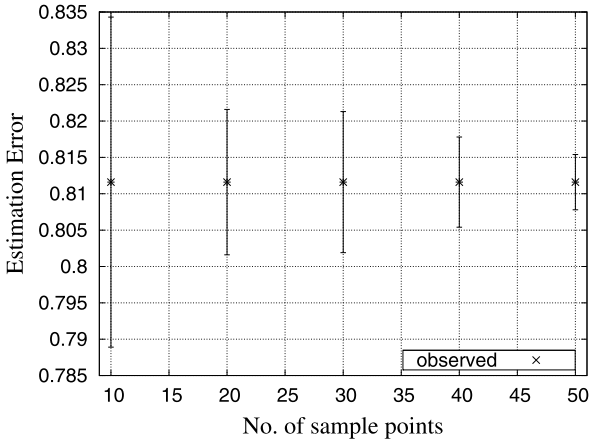


Fig. 20 Influence of "N" on the predictions [low-fidelity C_l]

than 4.8 %. The deviation can be further reduced by increasing the accuracy of the predictions by increasing the number of training sample points.

The constructed surrogate models have been coupled with the GA [34]. The parameters for controlling the ASO are summarised in Table 3. Each generation of the GA has 5 different individuals (i.e. airfoil geometries) and the ASO process is carried out in 500 generations. In total, 2,500 different airfoil geometries with different M are obtained and tested for maximum E. An optimised solution, which has an aerodynamic efficiency of $E = 80.326$, is obtained and converges around the 495th generation of the GA. The flow around the optimised airfoil geometry is solved in FLUENT using the corresponding flow properties as shown in Table 4. The CFD calculations show that the optimised airfoil geometry has $E = 77.106$, corresponding

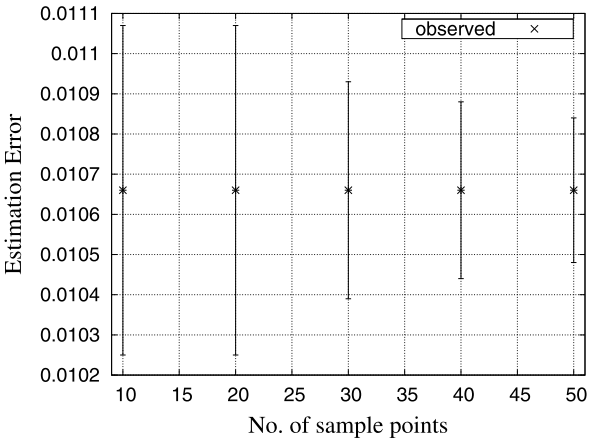


Fig. 21 Influence of "N" on the predictions [high-fidelity C_d]

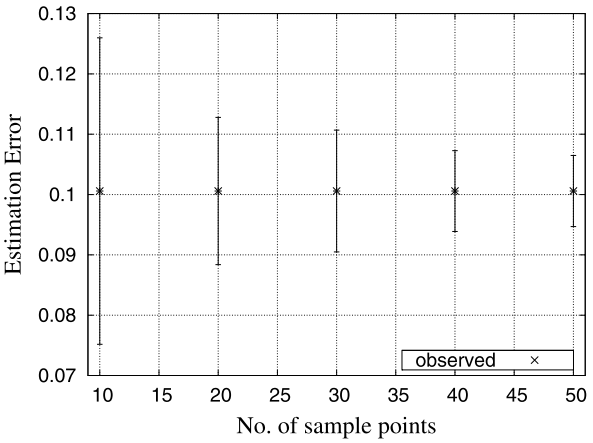


Fig. 22 Influence of "N" on the predictions [ΔC_l]

to a 4.1 % error. As already discussed, this value is less than the maximum expected error of 4.8 %. Despite the small prediction error, the obtained airfoil geometry is still better than the baseline shape, which has 67.015 for the flow conditions tabulated in Table 4, and other explored airfoil geometries. It can then be confirmed that the optimised geometry has 15.04 % improvement in E over the actual NACA 2411 at the specified flow conditions. The variation in the geometry, pressure and velocity distribution between the baseline airfoil and optimised airfoil are depicted in Figs. 24–29. It can be observed from these figures that the higher airflow acceleration at the suction side and higher positive C_p at the pressure side are the primary reasons for the optimised airfoil to have more E than its counterpart.

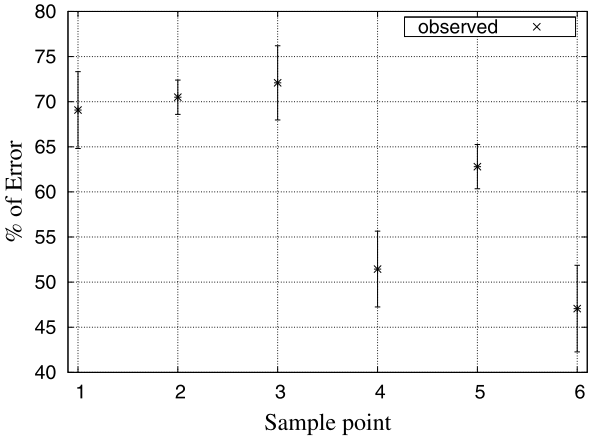


Fig. 23 % of error of E prediction

Table 3 GA parameters to control the evolution (ASO)

Parameters	Values
Number of parameters	11
Population size (*popsize*)	5
Crossover probability (p_c)	0.7
Mutation probability (p_m)	0.02

Table 4 Flow properties to solve the optimised geometry

Property	Value
Pressure (p)	101,325 N/m^2
Density (ρ)	1.1766 Kg/m^3
Temperature (T)	300 K
Mach (M)	0.331351
Velocity (V)	115.05
Re	7.337×10^6
I	0.022 %
μ	1.845×10^{-5} Kg/m s

The whole ASO problem is carried out in $0.341399E + 03$ sec (5.6 min) with a computer system which has 1.5 GB of DDR 2 RAM, 2.6 GHz of processor speed and 2 MB of L2 cache memory. If the actual CFD algorithm were to be employed for solving the flow during the optimisation, 156 days would have been required for the same computer system to obtain the optimised solution. This is calculated based on the time taken for a single CFD simulation (90 min approximately) during the data mining process. Clearly, applying the surrogate models in the place of actual CFD

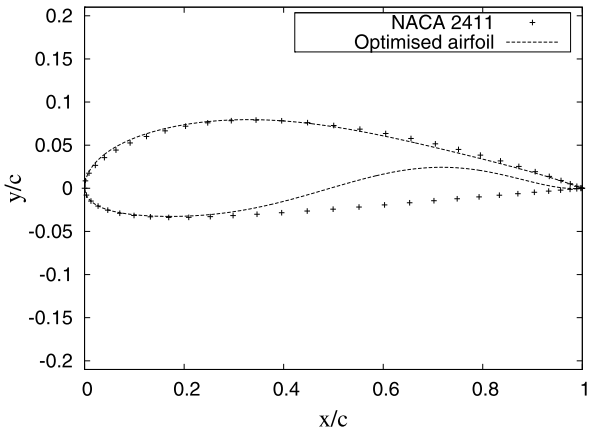


Fig. 24 Optimised geometry

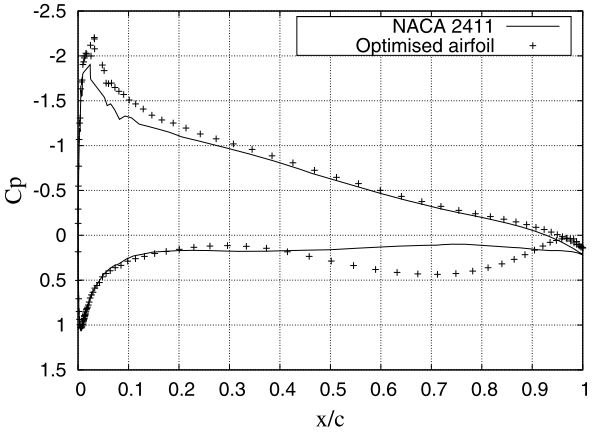


Fig. 25 C_p distribution

algorithms has drastically reduced the required computational time and resources to carry out an ASO problem.

The method of the parameterisation scheme is crucial for both surrogate model construction and optimiser, since its variables are used as the design and optimisation variables. The PARSEC parameterisation scheme is effective, since it offers flexibility in controlling the aerodynamic characteristics of the airfoil geometry with a minimum number of parameters. The distribution of sample points within the design space strongly influences the performance of the surrogate models. Hence, it is important to use a sampling plan that is able to explore the design space uniformly rather than just distributing the sample points in an arbitrary fashion. The Hammersley sequence sampling (HSS) technique has more uniformity and stratification properties for the given "N" irrespective of "n" of the problem. These properties make

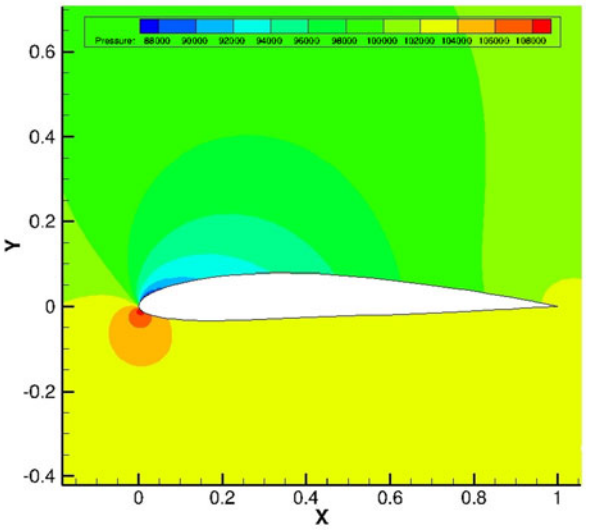


Fig. 26 Pressure distribution around the NACA 2411 airfoil

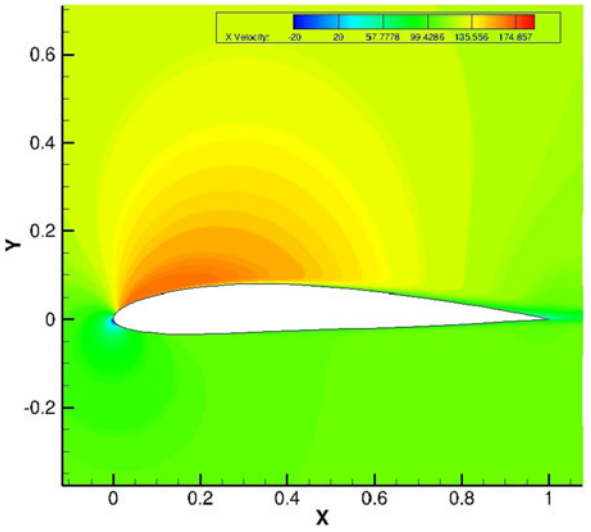


Fig. 27 Velocity distribution around the NACA 2411 airfoil

this algorithm suitable for problems where the evaluation of objective functions at one sample point is computationally more expensive. The statistically unbiased characteristics of the ordinary kriging (OK) approach enhance the ability and accuracy of the surrogate models in predicting response values at an unexplored space. The GA is observed to be more effective in exploring the search space. Since vari-

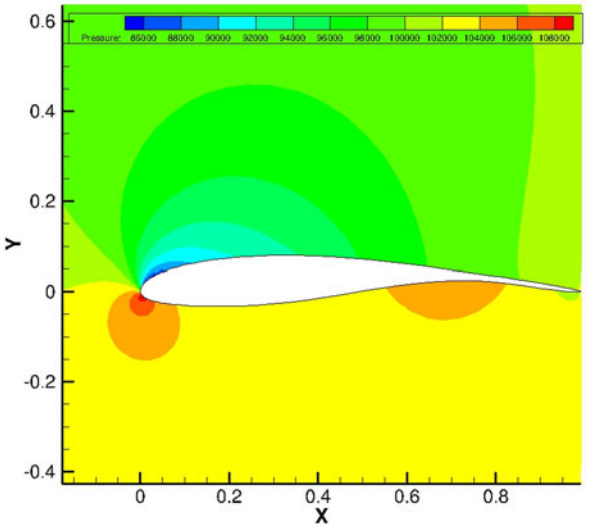


Fig. 28 Pressure distribution around the optimised airfoil

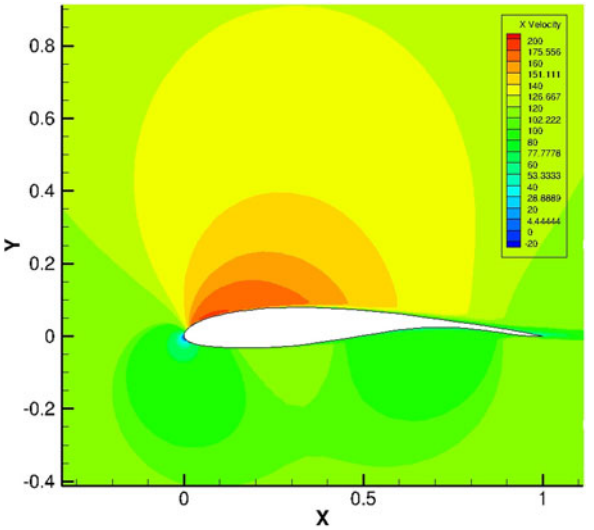


Fig. 29 Velocity distribution around the optimised airfoil

ability exists in all the generations of the GA, a huge number of desirable solutions to the defined problem are generated. Hence, this process can also be considered as a *data mining process* and can be further used for airfoil design and analysis.

4 Conclusions

This chapter may be summarised as follows. The chapter begins with a review of the basic activities involved in an aerodynamic shape optimisation problem. Next, it gives more information about the construction of the surrogate models for various aerodynamic functions and their application to the aerodynamic shape optimisation problems. The chapter is concluded with a discussion on the practical challenges involved in employing the surrogate models to aerodynamic shape optimisation problems.

Acknowledgements The authors would like to express their sincere gratitude to Dr. Raman Balu, Dean, School of Interdisciplinary Studies, NICHE, Tamilnadu, India for his valuable suggestions during the work.

References

1. Balu, R., Ulaganathan, S.: Optimum hierarchical Bezier parameterization of arbitrary curves and surfaces. In: 11th Annual CFD Symposium, 11–12 August 2009. Indian Institute of Science, Bangalore (2009)
2. Zang, T., Green, L.: Multidisciplinary design optimization techniques: implications and opportunities for fluid dynamics research. In: 30th AIAA Fluid Dynamics Conference, June 28–July 1, Norfolk, VA (1999)
3. Raymer, D.: Enhancing aircraft conceptual design using multidisciplinary optimization. Doctoral thesis, Kungliga Tekniska Hogskolan, Royal Institute of Technology, Sweden (1995). ISBN: 91-7283-259-2, May
4. Giuntia, A.: Aircraft multidisciplinary design optimization using design of experiments theory and response surface modeling methods. MAD Center Report: 97-05-01, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0203 (1997)
5. Duchaine, F., Morel, T., Gicquel, L.Y.M.: Computational-fluid-dynamics-based kriging optimization tool for aeronautical combustion chambers. AIAA, ISSN: 0001-1452, Vol. 47, No. 3, pp. 631–645, CERFACS, 42 Av. G. Coriolis, 31057 Toulouse, France (2009)
6. Xiong, J.T., Qiao, Z.D., Han, Z.H.: Aerodynamic shape optimization of transonic airfoil and wing using response surface methodology. In: 25th International Congress of the Aeronautical Sciences, ICAS 2006-2.1.3, 3–8 September 2006, Hamburg, Germany (2006)
7. Laurenceau, J., Sagaut, P.: Building efficient response surfaces of aerodynamic functions with kriging and cokriging. CERFACS, Toulouse, 31057, France, January 24 (2008)
8. Sobieczky, H.: Parametric airfoils and wings. In: Notes on Numerical Fluid Mechanics, pp. 71–78. Vieweg, Wiesbaden (1998)
9. Avinash, G.S., Anil Lal, S.: Inverse Design of Airfoil Using Vortex Element Method. Department of Mechanical Engineering, College of Engineering, Thiruvananthapuram, Kerala, India (2010)
10. Nadarajah, S., Castonguay, P.: Effect of shape parameterization on aerodynamic shape optimization. In: 45th AIAA Aerospace Science Meeting and Exhibit, Reno, Nevada, January 8–11 (2007)
11. Selvakumar, U., Mukesh, P.R.: Aerodynamic shape optimization using computer mapping of natural evolution process. In: 2010 2nd International Conference on Computer Engineering and Technology (ICCET), vol. 5, pp. 367–371 (2010). ISBN: 978-1-4244-6347-3
12. Hajek, J.: Parameterization of airfoils and its application in aerodynamic optimization. In: WDS'07 Proceedings of Contributed Papers, Part I, pp. 233–240 (2007). ISBN:978-80-7378-023-4

13. Wang, G., Shan, S.: Sampling strategies for computer experiments: design and analysis. Int. J. Reliab. Appl. **2**(3), 209–240 (2001)
14. Giunta, A.A., Wojtkiewicz, S.F. Jr., Eldred, M.S.: Overview of modern design of experiments methods for computational simulations. Sandia National Laboratories, Albuquerque, NM, AIAA 0649 (2003)
15. Kalagnanam, J.R., Diwekar, U.M.: An efficient sampling technique for off-line quality control. American Statistical Association and the American Society for Quality Control, vol. 39, No. 3, August (1997)
16. Jouhaud, J.-C., Sagaut, P., Montagnac, M., Laurenceau, J.: A surrogate-model based multidisciplinary shape optimization method with application to a 2D subsonic airfoil. Comput. Fluids **36**, 520–529 (2007)
17. Andersson, J.: A survey of multiobjective optimization in engineering design. Department of Mechanical Engineering, Linkoping University, Sweden, Technical report: LiTH-IKP-R-1097
18. Alba, E., Cotta, C.: Evolutionary algorithms, Dept. Lenguajes y Ciencias de la Computacion, ETSI Informatica, Universidad de Malaga, Campus de Teatinos, 29016, Malaga, Spain, February 19 (2004)
19. Holland, J.: Adaption in Natural and Artificial Systems. MIT Press, Cambridge (1975)
20. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Learning. Addison-Wesley Longman, Inc., Boston (1989)
21. Balu, R.: Natural evolution as a process of optimisation. Aerodynamics Research and Development Division, VSSC, ISRO, India (1999)
22. Miller, B.L., Goldberg, D.E.: Genetic algorithms, tournament selection, and the effects of noise. Complex Systems **9**, 193–212 (1995)
23. Alam, M.N.: Optimization of VLSI circuit by genetic algorithms. Lecture Notes, University of Vassa (2006)
24. Kroo, I.: Applied Aerodynamics, Desktop Aerodynamics. P.O. Box 20384, Stanford, CA 94309, January (2007)
25. FLUENT: Fluent 6.1 User's Guide, Fluent Inc. 25/01 (2003)
26. Spalart, P.R., Allmaras, S.R.: A one-equation turbulence model for aerodynamic flow. AIAA Paper 92-0439 (1992)
27. Wilcox, D.C.: Turbulence modelling for CFD. DCW Industries Inc, La Canada, California (1993)
28. Liang, Y., Cheng, X.-q., Li, Z.-n., Xiang, J.-w.: Robust multi-objective wing design optimisation via CFD approximation model. Eng. Appl. Comput. Fluid Mech. **5**(2), 286–300 (2011)
29. Greschner, B., Yu, C., Zheng, S., Zhuang, M., Wang, Z.J., Thiele, F.: Knowledge based airfoil aerodynamic and aeroacoustic design. AIAA, pp. 1–11, May (2005)
30. Internet Material: Information on Y plus wall distance estimation. www.cfd-online.com, Accessed on August 12, 2011
31. Kurganov, A., Tadmor, E.: New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations. J. Comput. Phys. **160**, 214–282 (2010)
32. Metodiev, K.K.: Euler computations of supercritical airfoils. Space Research Institute, Bulgarian Academy of Sciences, Sofia, Bulgaria
33. Sheldahl, R.E., Klimas, P.C.: Aerodynamic characteristics of seven airfoil sections through 180 degrees angle of attack for use in aerodynamic analysis of vertical axis wind turbines. SAND80-2114, Sandia National Laboratories, Albuquerque, NM, March (1981)
34. Carroll, D.L.: FORTRAN code for Genetic Algorithm, Version 1.7a. CU Aerospace, 2004 South Wright Street Extended, Urbana, IL 61802 (2001)

Knowledge-Based Surrogate Modeling in Engineering Design Optimization

Qian Xu, Erich Wehrle, and Horst Baier

Abstract Simulations and numerical experiments of engineering problems are often expensive, which may restrict sensitivity analysis and design optimization. Surrogate modeling methodologies are currently being studied to construct approximation models of system responses based on a limited number of the expensive evaluations. The use of surrogate models allows more efficient exploration and exploitation of the system. However, the curse of dimensionality is still an obstacle for large and complex engineering design problems. The required number of high-fidelity evaluations becomes tremendously large in a high-dimensional space. Therefore, it is advisable to adopt knowledge-based surrogate modeling in engineering design optimization. With engineering insight into the system, a high-dimensional design space can be intelligently mapped into system properties, so that better choices of inputs, outputs, and function formulations can be made for surrogate modeling. This chapter covers the methods of embedding engineering knowledge in surrogate modeling for structural mechanical systems and provides application examples in the field of aerospace engineering.

Keywords Surrogate model · Kriging · Structural optimization · Surrogate-based design optimization · Knowledge-based surrogate modeling

1 Introduction and Overview of Surrogate Modeling in Engineering Optimization

Engineering design optimization finds the best available values of design variables to maximize or minimize design objective function(s) while satisfying all specified

Q. Xu (✉) · E. Wehrle · H. Baier
Institute of Lightweight Structures, Technische Universität München, Boltzmannstr. 15, 85748 Garching, Germany
e-mail: q.xu@llb.mw.tum.de

E. Wehrle
e-mail: wehrle@tum.de

H. Baier
e-mail: baier@llb.mw.tum.de

S. Koziel, L. Leifsson (eds.), *Surrogate-Based Modeling and Optimization*,
DOI 10.1007/978-1-4614-7551-4_13,

constraints in engineering problems [4]. It usually requires a number of evaluations of the objective and constraint functions, which can be complex and time consuming. Sources of complexity include high-dimensional and nonconvex design spaces, highly nonlinear system responses, multidisciplinary design problems, etc. High-fidelity evaluations for many engineering design problems, i.e., simulations with finite element analysis (FEA) and computational fluid dynamics (CFD) of complex engineering systems, can take many hours or even days to complete. For parameter and sensitivity studies many evaluations are required and, further, design optimization can be limited by extensive computational effort. For engineering design optimization problems for which system equations and gradient information are expensive to obtain, it is worth using surrogate models to accelerate the engineering design process. Surrogate modeling involves generating system models, which are cheaper to evaluate. In this chapter, system response surface approximation is explained.

Surrogate models are constructed after a design of experiments (DoE) [19] with a number of high-fidelity evaluations carried out. The process involves two basic steps: sampling and surrogate model construction, both of which can be affected by the curse of dimensionality resulting from a large number of design variables [17].

1. Sampling: The sample size required in DoE grows tremendously with the increasing dimension of the design space, i.e., at least $\frac{(k+1)(k+2)}{2}$ samples are required to perform a full quadratic polynomial regression in k-dimensional design space.
2. Surrogate model construction: High dimensionality requires more coefficients and parameters to be determined to best fit the samples. Matrix operations such as matrix inverse or Cholesky and LU decomposition have to be performed for large matrices in least-squares estimation. Large numerical errors can occur when a poor combination of regression model and design sites is formulated and the matrices in the regression model become ill conditioned.

Parallel computation can alleviate some problems associated with the curse of dimensionality, but it remains a restricting factor for the use of surrogate modeling. Studies on surrogate modeling strategies are oriented to reduce computational effort without significantly compromising model accuracy [16]. Strategies studied in recent years include but are not limited to the following.

1. Sequential design space reduction and expected improvement:
   Giunta [9] estimates the improvement of surrogate models, based on which the size of the trust region is updated and the design space is reduced. Expected improvement as an infill criterion is described by Forrester [8], according to which additional sample points (infill points) are generated to refine surrogate models. Such strategies risk losing characteristics of system responses, especially for highly nonlinear systems [21].
2. Gradient information:
   For selected sample sites, derivative information is included to increase approximation accuracy. Gradient-enhanced and Hessian-enhanced surrogate modeling

show the possibility of building more accurate predictions [8]. However, analytical or semi-analytical gradients are not always available, and numerical gradients are expensive to calculate.

3. Reduced basis functions:
   Substitutes of full-polynomial terms are utilized to reduce the number of coefficients, thus reducing the required number of samples. Alvarez et al. [2] propose a tree-searching algorithm and use genetic programming methodology to generate the basis terms that fit best to the sample points. Such methods show considerable potential in reducing sample points and increasing accuracy for many design problems. However, candidate basis functions or operations are required in these methods. Further improvements could be expected by providing a better choice of candidates considering engineering knowledge.
4. Kernel-based method and space mapping:
   Instead of modeling the design space, a mapped space is used. Baudat [6] uses a kernel-based method, which extracts a relevant data set into a feature space according to geometrical considerations. Bandler [5] introduces space mapping procedures to iteratively update and optimize surrogates based on a physical-based coarse model, in which engineering experience is adopted.

Combinations of the techniques above may also be used to further enhance surrogate modeling.

In this chapter, knowledge-based surrogate modeling will be described. The focus will be on choosing surrogate modeling factors such as inputs, outputs, and function formulations based on knowledge of an engineering system. The implementation of knowledge-based surrogate modeling is realized by embedding terms in the surrogate functions, which are mapped from the design space and are able to represent the system properties. With those terms fewer surrogate parameters need to be determined, and the relationship to be approximated becomes more directly related or even linearly related. Application examples in aerospace engineering are given to elaborate the implementation of knowledge-based surrogate modeling.

2 Surrogate Modeling with Latin Hypercube Sampling and Kriging

There are a variety of methods used for sampling and construction of surrogate models. Typical sampling methods are Latin hypercube sampling (LHS), central composite designs, Box-Behnken designs, orthogonal arrays, and Hammersley sequences [18]. Surrogate modeling methods include, e.g., polynomial regression, kriging, radial basis functions (RBF) [15], artificial neural networks (ANN), and support vector regression (SVR) [7]. In this section, the LHS method and the kriging modeling method are briefly described and extended to perform knowledge-based surrogate modeling.

Latin Hypercube Sampling It is important to control the density of sample points so that the corresponding amount of evaluations is affordable and still maintains good distribution in the design space. LHS is a high-dimensional extension of Latin square sampling. In two-dimensional space, samples are distributed in square grids of the design domain with only one sample in each column and each row. For a higher-dimensional space, the range of each design variable is divided into the required number of intervals. In each interval of each design variable, only one sample point is allowed. This feature benefits design space mapping strategies, because duplicate points can be avoided and, therefore, sample points can be maximally utilized.

Kriging Kriging, which is widely used, is a combination of polynomial regression and Gaussian stochastic processes. The regression model fits the samples according to the rule of least-squares estimation. The correlation model adjusts the prediction error by using maximum likelihood estimation. Kriging is formulated as

$$\hat{y} = Y(\mathbf{x}) + Z(\mathbf{x}) = \overbrace{\sum_{i=0}^{n} \beta_i f_i(\mathbf{x})}^{\text{regression}} + \underbrace{Z(\mathbf{x})}_{\text{correlation}}, \tag{1}$$

where $Y(\mathbf{x}) = \sum_{i=0}^{n} \beta_i f_i(\mathbf{x})$ is the regression model, which usually adopts polynomials up to the second order to represent the global trend of the sample points. $Z(\mathbf{x})$ is the correlation model, which is a Gaussian process with mean value 0 and covariance σ. The correlation model is given by

$$\mathrm{Cov}\big(Z(\mathbf{x}_j), Z(\mathbf{x}_k)\big) = \sigma^2 R_{jk}(\boldsymbol{\theta}, \mathbf{x}_j, \mathbf{x}_k), \tag{2}$$

where R_{jk} is the Gaussian correlation function on the p-dimensional design space

$$R_{jk}(\boldsymbol{\theta}, \mathbf{x}_j, \mathbf{x}_k) = \prod_{i=1}^{p} e^{-\theta_i (|\mathbf{x}_{ji} - \mathbf{x}_{ki}|^2)}. \tag{3}$$

In Eqs. (2) and (3), $\boldsymbol{\theta}$ is the coefficient vector of the correlation model. The optimal coefficients $\boldsymbol{\theta}^*$ are found for maximum likelihood estimation. It is important to choose proper formulations of the regression functions which represent the system behavior as precisely as possible. Higher-order polynomial regressions have the capability of approximating more complex responses, but they require more sample points to determine the polynomial coefficients. By properly selecting the polynomial orders and mixed terms of design variables using knowledge of a system, the number of coefficients in regression functions can be greatly reduced.

The relative root mean square error $\varepsilon_{\mathrm{rrms}}$ is used for comparison of surrogate models. It is defined as the root mean square error normalized by the maximal response value,

$$\varepsilon_{\mathrm{rrms}} = \frac{1}{\max(y)} \sqrt{\frac{\sum_{i=0}^{n} [\hat{y}(x_i) - y(x_i)]^2}{n}}. \tag{4}$$

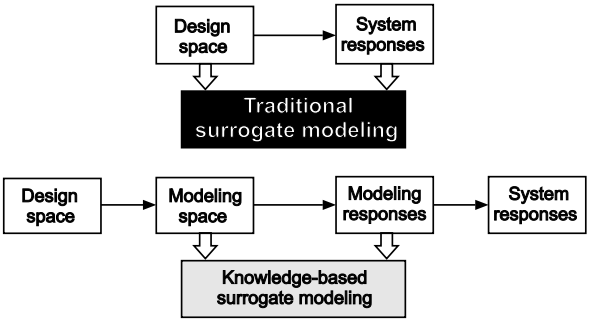


Fig. 1 Traditional metamodel and knowledge-based metamodel

3 Knowledge-Based Surrogate Model

The efficiency and accuracy of the traditional surrogate models can be improved by considering the physical properties of the engineering system. These systems are well understood by engineers and, therefore, do not have an unknown black-box nature. Figure 1 compares traditional and knowledge-based surrogate modeling methods. The traditional surrogate modeling process is performed assuming the system as a black box. The information from the design space and system responses is taken to construct a surrogate model without considering engineering understanding of the system. However, in knowledge-based surrogate modeling, engineering knowledge is considered in the surrogate modeling process. The design space is mapped into a modeling space, which is composed of structural mechanical properties of the system. The system responses are replaced by modeling responses, which can be more accurately approximated.

This section first explains the implementation of the knowledge-based surrogate modeling method. An introductory example is then given for more detailed description of the method and its advantages.

3.1 Implementation of Knowledge-Based Surrogate Modeling with Kriging

Knowledge-based kriging is constructed by considering the mechanical system properties and is formulated as

$$\hat{q}(y) = Y\big(\mathbf{p}(\mathbf{x})\big) + Z\big(\mathbf{p}(\mathbf{x})\big) = \overbrace{\sum_{i=0}^{n} \beta_i f_i\big(\mathbf{p}(\mathbf{x})\big)}^{\text{regression}} + \underbrace{Z\big(\mathbf{p}(\mathbf{x})\big)}_{\text{correlation}}, \tag{5}$$

where $\mathbf{p}(\mathbf{x})$ is used to form polynomial terms instead of $\mathbf{x}$. $\mathbf{p}(\mathbf{x})$ is an algebraic combination of the design variables that represents the structural mechanical properties. $q(y)$ is transformed from y using engineering knowledge to form a relationship that

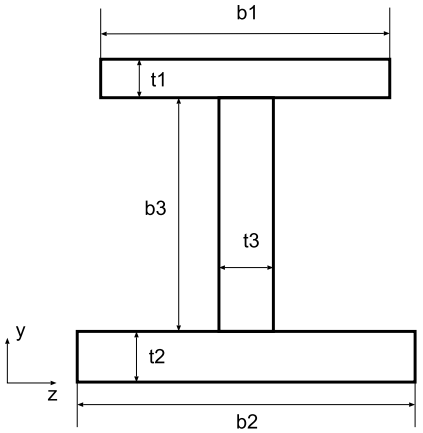


Fig. 2 Cross section of a cantilever with design variables

can be more accurately represented by kriging. The use of knowledge-based kriging has the following advantages:

- Fewer coefficients β_i are determined: Since only physically sensible terms are considered, the number of mixed terms in the regression function can be reduced.
- A higher approximation accuracy can be achieved: Compared with the original system responses, the modeling outputs $q(y)$ are more directly related or even linearly related to the modeling inputs $\mathbf{p}(\mathbf{x})$.

3.2 Introductory Example of Knowledge-Based Surrogate Modeling

The design problem of a cantilever is shown as an introductory example of utilizing engineering knowledge for surrogate modeling. The cantilever has an I-shaped cross section (see Fig. 2), the geometric parameters of which are to be determined to minimize the structural mass, subject to a number of structural constraints. The cantilever is loaded with force in the y-direction at the tip and is clamped at the opposite end. The responses to be analyzed are the bending stress σ_b, the shear stress τ, the tip deflection δ, the failure force for twist buckling F_{cr}, and the first resonance frequency of bending f_b. The design problem is formulated as follows:

$$\begin{aligned} &\text{minimize} && m(\mathbf{x}) \\ &\text{subject to} && \sigma_b(\mathbf{x}) \le \sigma_{\mathrm{crit}} \\ &&& \tau(\mathbf{x}) \le \tau_{\mathrm{crit}} \\ &&& \delta(\mathbf{x}) \le \delta_{\mathrm{crit}} \\ &&& -F_{\mathrm{cr}}(\mathbf{x}) \le -F_{\mathrm{load}} \\ &&& -f_b(\mathbf{x}) \le -f_0 \\ &\text{and} && \mathbf{x}_l \le \mathbf{x} \le \mathbf{x}_u, \end{aligned} \tag{6}$$

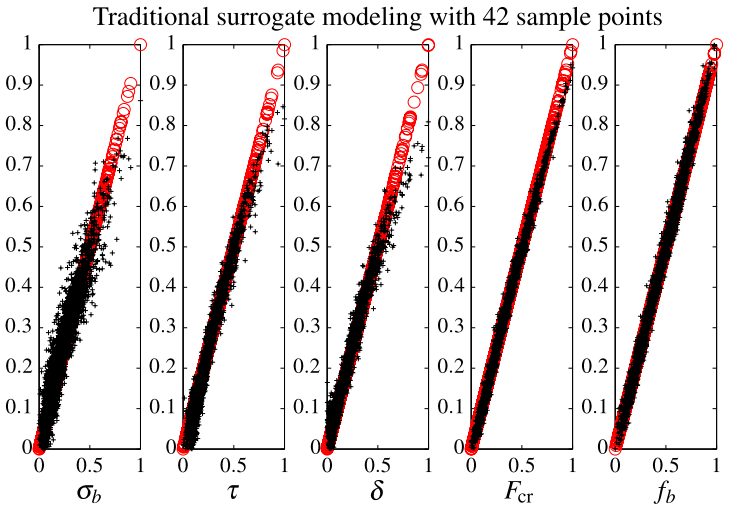


Fig. 3 Traditional surrogate modeling results of the cantilever with I-shaped cross section. *Circle* (o): system responses, *cross* (+): surrogate function values

where $\mathbf{x} = \{b_1, b_2, b_3, t_1, t_2, t_3\}^T$ (see Fig. 2), $\mathbf{x}_l$ and $\mathbf{x}_u$ refer to the lower and upper boundaries of the design variables, and m refers to the mass of the structure. σ_{crit}, τ_{crit}, δ_{crit}, F_{load}, and f_0 are constant values in the design criterion. Surrogate models for σ_b, τ, δ, F_{cr}, and f_b are built with respect to $\mathbf{x}$.

For a six-dimensional design space, at least 28 sample points are required for a quadratic regression. Considering a typical oversampling factor of 1.5, 42 sample points are generated by LHS. The surrogate models are constructed with kriging. The quality of the surrogate model is tested with 3,000 high-fidelity system responses. The results are shown in Fig. 3: Shown from left to right are normalized values of σ_b, τ, δ, F_{cr}, and f_b, respectively. The consistency of system responses and surrogate function values represent the quality of surrogate modeling. As can be seen, the quality of the surrogate models can be further improved, especially for the bending stress and tip deflection.

With a knowledge of structural mechanics, it is known that the system responses are dependent on the geometrical and mechanical properties, such as the cross-sectional area A, the position of the center of the cross section, the area moment of inertia I_z and I_y, etc. It is also known that the bending stress is inversely proportional to the area moment of inertia I_z, and directly proportional to the distance from the geometrical center $h_{\max}$. The system structural responses and the respective geometrical properties are shown in Table 1.

The geometrical properties $h_{\max}$, I_y, I_z, and A are directly related to the system responses. With these listed terms used as $\mathbf{p}(\mathbf{x})$ in the kriging regression functions, fewer coefficients are to be determined and fewer sample points are required. The results of knowledge-based surrogate modeling are shown in Fig. 4. A comparison of the surrogate modeling results is shown in Table 2. n_{samp} represents the number

Table 1 The structural responses and the respective geometrical properties

Structural responses	σ_b	τ	δ	F_{cr}	f_b
Geometrical properties	$h_{\max}, \frac{1}{I_z}$	$\frac{1}{A}$	$\frac{1}{I_z}$	I_y, I_z	$\sqrt{\frac{I_z}{A}}$

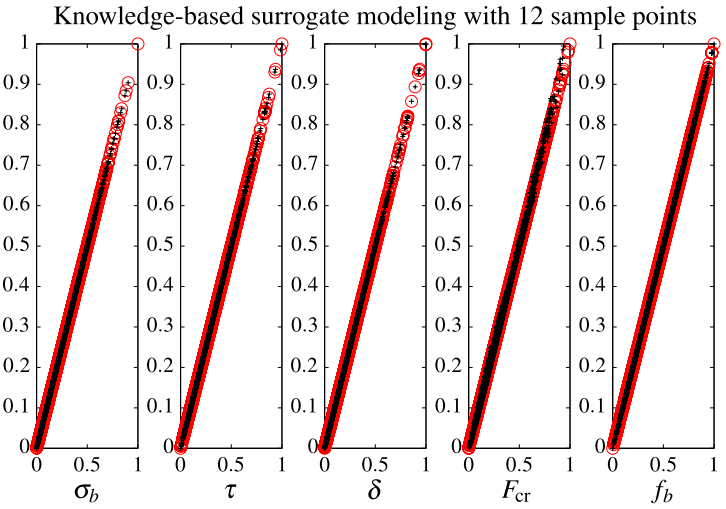


Fig. 4 Knowledge-based surrogate modeling results of the cantilever with I-shaped cross section. *Circle* (o): system responses, *cross* (+): surrogate function values

Table 2 Results of the traditional surrogate modeling method and the knowledge-based surrogate modeling method for the cantilever with I-shaped cross section

	n_{samp}	$\varepsilon_{\mathrm{rrms}}(\sigma_b)$	$\varepsilon_{\mathrm{rrms}}(\tau)$	$\varepsilon_{\mathrm{rrms}}(\delta)$	$\varepsilon_{\mathrm{rrms}}(F_{\mathrm{cr}})$	$\varepsilon_{\mathrm{rrms}}(f_b)$
Traditional surrogate model	42	5.42 %	1.91 %	4.36 %	1.12 %	1.12 %
Knowledge-based surrogate model	12	0.00 %	0.00 %	0.00 %	1.02 %	0.00 %

of sample points used in surrogate modeling. Compared with the traditional surrogate modeling approach, the knowledge-based surrogate modeling method not only saves sample points but also results in a much higher accuracy. For complex mechanical structures, the mechanical properties would not be directly found as in the cantilever example shown here. However, one can obtain clues from engineering knowledge and experience.

4 Knowledge-Based Terms for Structural Systems

In this section, the construction of a knowledge-based modeling space for structural systems is described. The equation of motion is considered here:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{F}, \tag{7}$$

where $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ are the global mass, damping, and stiffness matrices, respectively. $\mathbf{K}$ and $\mathbf{M}$ are combinations of design variables that most directly describe the relation between geometric parameters and system responses. The global matrices $\mathbf{K}$ and $\mathbf{M}$ are assembled from the element matrices $\mathbf{K}^e$ and $\mathbf{M}^e$, which can be explicitly derived from the design variables.

Beam and shell elements are frequently employed element types in aerospace engineering. The element matrices will be extracted to assist in forming the modeling space. Besides traditional metallic materials, composite materials have also been commonly applied in the aerospace industry, the mechanical properties of which will also be extracted to assist knowledge-based surrogate modeling.

4.1 Knowledge-Based Terms for Beam Elements

For beam elements, the geometric parameters of the cross section are considered in engineering design. The complete stiffness matrix for the 12 degree-of-freedom beam element is given by [10]

$$\mathbf{K}^e = \begin{bmatrix} \mathbf{K}^e_{11} & \mathbf{K}^e_{12} \\ sym & \mathbf{K}^e_{22} \end{bmatrix}, \tag{8}$$

in which

$$\mathbf{K}^e_{11} = \begin{bmatrix} \frac{EA}{l} & 0 & 0 & 0 & 0 & 0 \\ & \frac{12EI_z}{l^3\Psi_y} & 0 & 0 & 0 & \frac{6EI_z}{l^2\Psi_y} \\ & & \frac{12EI_y}{l^3\Psi_z} & 0 & -\frac{6EI_y}{l^2\Psi_z} & 0 \\ & & & \frac{GJ}{l} & 0 & 0 \\ & sym & & & \frac{\Psi'_z EI_y}{l} & 0 \\ & & & & & \frac{\Psi'_y EI_z}{l} \end{bmatrix}, \tag{9}$$

where E, G, A, and l represent Young's modulus, the shear modulus, the cross-sectional area, and the length of the beam element, respectively. Refer to [10] for details of $\mathbf{K}^e_{12}$, $\mathbf{K}^e_{22}$ and the substitutions Ψ_y, Ψ'_y, Ψ_z, and Ψ'_z. In the stiffness matrix $\mathbf{K}^e$, the terms EA, $\frac{EI}{\Psi}$, $EI\Psi'$, and GJ are combinations of design variables that represent mechanical properties of axial, bending, shear, and torsional stiffnesses, respectively. Similarly, in the element mass matrix $\mathbf{M}^e$, terms such as $\frac{I}{A}$ and $\frac{J}{A}$ are combinations of design variables that represent translational and rotational inertia. These terms can be used to form the modeling space. Based on the responses to be approximated, different combinations of mechanical properties can be selected

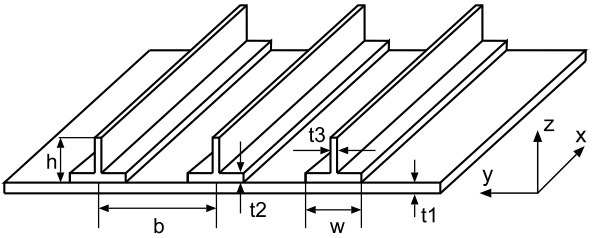


Fig. 5 Stiffened aircraft wing skin. The stringers are constructed with beam elements, and the skin is constructed with shell elements

to represent the system behavior under particular load cases. The combinations are used as the basis for $\mathbf{p}(\mathbf{x})$ of the regression model in kriging for knowledge-based surrogate modeling.

4.2 Knowledge-Based Terms for Shell Elements

For shell elements, the thicknesses are often considered as design variables in engineering sizing optimization. To prevent local buckling of the shell structures, they are usually stiffened at certain distances, which may also be considered as design variables. t, b, and l represent the thickness, and the horizontal and the vertical sizes, respectively. The membrane stiffness matrix $\mathbf{K}_M^e$, bending stiffness matrix $\mathbf{K}_B^e$, and shear stiffness matrix $\mathbf{K}_S^e$ of a shell element are given by [10]

$$\begin{aligned} \mathbf{K}_M^e &= \frac{t}{4A}\begin{bmatrix} \mathbf{P} & \mathbf{R} \\ sym & \mathbf{Q} \end{bmatrix}, \qquad \mathbf{K}_B^e = \frac{t^3}{48A}\begin{bmatrix} \mathbf{Q} & -\mathbf{R}^T \\ sym & \mathbf{P} \end{bmatrix}, \\ \mathbf{K}_S^e &= \frac{t}{4A}\begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} & \mathbf{S}_{13} \\ & \mathbf{S}_{22} & \mathbf{S}_{23} \\ sym & & \mathbf{S}_{33} \end{bmatrix}, \end{aligned} \tag{10}$$

where $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{R}$, and $\mathbf{S}$ are position-dependent matrices, which are composed of quadratic terms of the horizontal and the vertical parameters b and l. A represents the shell face area, which is directly proportional to bl. As a result, terms involving t, t^3, $\frac{b}{l}$, and $\frac{l}{b}$, as well as their production should be considered in the regression model of kriging for knowledge-based surrogate modeling.

4.2.1 Demonstrative Example: Knowledge-Based Surrogate Modeling in Design of Aircraft Wing Skin

To describe how the above-mentioned mechanical properties are used in constructing knowledge-based surrogate models, we give a structural design example. Consider a piece of stiffened aircraft wing skin (see Fig. 5), for which the stringers are constructed with beam elements and the skin is constructed with shell elements. The wing is clamped at one end, representing the connection with the fuselage, and is

free at the other end. Distributed pressure is loaded on the skin surface. The design variables are as shown in Fig. 5: the thickness of the skin t_1, the flange t_2, and the web t_3, width of the flange w, height of the web h, and the horizontal distance b between stringers. The maximum bending stress σ_b, the tip deflection δ, and the first two bending frequencies f_{b1} and f_{b2} are responses of the structural system that will constrain the weight minimization design. This is formulated as

$$\begin{aligned} &\text{minimize} && m(\mathbf{x}) \\ &\text{subject to} && \sigma_b(\mathbf{x}) \le \sigma_{\text{crit}} \\ &&& \delta(\mathbf{x}) \le \delta_{\text{crit}} \\ &&& -f_{b1}(\mathbf{x}) \le -f_1 \\ &&& -f_{b2}(\mathbf{x}) \le -f_2 \\ &\text{and} && \mathbf{x}_l \le \mathbf{x} \le \mathbf{x}_u, \end{aligned} \tag{11}$$

where $\mathbf{x} = \{t_1, t_2, t_3, w, h, b\}^T$ (see Fig. 5). Considering the mechanical properties of the structure, the following attributes are used in the regression function of kriging:

- Area moment of inertia of the skin I_{skn} and the stringer I_{str} around the y-axis. Since the skin and the stringers are combined as a whole structure, the area moment of inertia is not calculated in their local coordinates but in the global coordinate system. The summation $I_{\text{all}} = I_{\text{skn}} + 3I_{\text{str}}$ is also considered for the global behavior of the structure.
- The geometrical maximum distance to the center of the cross section, $h_{\max}$, which is also a physically sensible combination of the design variables.
- The cross-sectional area of the skin A_{skn} and the stringer A_{str}. The summation $A_{\text{all}} = A_{\text{skn}} + 3A_{\text{str}}$ is also an important property of the system global behavior, such as vibration frequencies [12].

These terms are combinations of design variables and used as the knowledge-based terms $\mathbf{p}(\mathbf{x})$ in kriging to perform knowledge-based surrogate modeling. By including these properties in surrogate modeling, the dimension and complexity of the problem are reduced. A sensitivity study of the system responses with respect to the knowledge-based terms has been performed (see Fig. 6). As can be seen, a quasi one-dimensional linear relationship is shown between

- the maximum bending stress σ_b and $\frac{h_{\max}}{I_{\text{all}}}$,
- the tip deflection δ and $\frac{1}{I_{\text{all}}}$,
- the bending frequencies and $\sqrt{\frac{I_{\text{all}}}{A_{\text{all}}}}$.

It is shown in Fig. 6 that by adopting knowledge-based terms, the responses to be approximated are reduced from multidimensional nonlinear cases to one-dimensional linear cases. A comparison of the traditional surrogate modeling method and the knowledge-based surrogate modeling method is shown in Table 3. It can be seen that the knowledge-based surrogate modeling method uses only 12

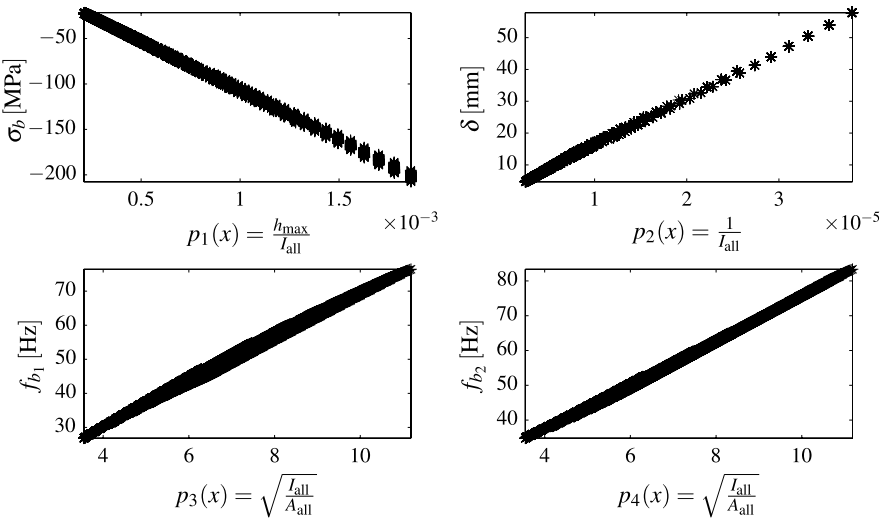


Fig. 6 A sensitivity study of the system responses to the knowledge-based terms

Table 3 Results of the traditional surrogate modeling method and the knowledge-based surrogate modeling method for the stiffened aircraft wing skin

	n_{samp}	$\varepsilon_{\text{rrms}}(\sigma_b)$	$\varepsilon_{\text{rrms}}(\delta)$	$\varepsilon_{\text{rrms}}(f_{b1})$	$\varepsilon_{\text{rrms}}(f_{b2})$
Traditional surrogate model	50	3.83 %	3.01 %	1.99 %	2.02 %
Knowledge-based surrogate model	12	0.71 %	0.56 %	0.38 %	0.96 %

high-fidelity analyses for this problem. The relative root mean square error of the approximations are calculated on 1,000 test points of the system responses; the test points are used to verify the precision of the surrogate models. In complex engineering problems, a large number of test points will not be provided, and the accuracy of surrogate models must be verified using other methods, such as the Bayesian information criterion (BIC), cross-validation [11], etc. As can be seen, the knowledge-based surrogate modeling method has benefited from the lower-dimensional relation and requires much fewer samples to get higher precision than the traditional surrogate modeling method.

4.3 Knowledge-Based Terms for Laminated Fiber Composite Structures

Composite materials have increasingly widespread application in aerospace structures. The system behavior of composite materials with respect to the design parameters is generally highly nonlinear. Therefore, the adoption of engineering knowl-

edge becomes even more important. As an example in this section, laminated fiber composites are applied to describe how engineering knowledge of composites is extracted to assist in surrogate modeling. For laminated fiber composites, usually the stresses and strains in the laminate are to be determined for known loads. The system equation is written as [1]

$$\begin{Bmatrix} \boldsymbol{\varepsilon}^0 \\ \mathbf{k} \end{Bmatrix} = \begin{bmatrix} \mathbf{A}^* & \mathbf{B}^* \\ \mathbf{B}^* & \mathbf{D}^* \end{bmatrix} \begin{Bmatrix} \mathbf{N} \\ \mathbf{M} \end{Bmatrix}, \tag{12}$$

where $\mathbf{N}$ and $\mathbf{M}$ are vectors representing the forces and moments acting on a laminate cross section. $\boldsymbol{\varepsilon}^0$ and $\mathbf{k}$ are the midplane strains and the plate curvatures, respectively. The matrices $\mathbf{A}^*$, $\mathbf{B}^*$, $\mathbf{D}^*$ are inverted from the matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{D}$, which are the extensional stiffness matrix, coupling stiffness matrix, and bending stiffness matrix, respectively [1]. Thin laminates can be considered as two-dimensional cases. There are six independent matrix elements for each matrix, which are derived from the lamina stiffness matrix $\bar{\mathbf{Q}}$ in the global coordinate system. The stresses on a lamina are then calculated as

$$\boldsymbol{\sigma}_k = [\bar{\mathbf{Q}}]_k \boldsymbol{\varepsilon}^0 + z[\bar{\mathbf{Q}}]_k \mathbf{k}. \tag{13}$$

The matrices $\mathbf{A}^*$, $\mathbf{B}^*$, $\mathbf{D}^*$, and $\bar{\mathbf{Q}}$ allow the laminate stresses and strains to be solved directly. Using elements of such matrices instead of the original design variables can generate more accurate approximations of the system responses in surrogate modeling. The system responses such as stresses, strains, and deflection are approximated with linear or quadratic relations of these matrix elements.

4.3.1 Demonstrative Example: Knowledge-Based Surrogate Modeling in Fiber Angle Design of Stiffened Composite Plate

The implementation of knowledge-based surrogate modeling is described here with a fiber angle design. Knowledge-based surrogate models for the fiber composite structural systems can be further used in engineering designs, such as stacking sequence optimization [20]. The structure in the example comprises laminated skin and stiffeners, both of which are composed of laminated carbon fiber reinforced polymer (CFRP) (see Fig. 7). The skin and the flanges have the same fiber angle design of 40 plies, and the webs have 60 plies. All laminates have a symmetric stacking sequence with respect to their middle planes. As a result, 50 fiber angles (20 for the skin and 30 for the webs) are to be determined. The fiber angle design is subjected to strength, stiffness, and stability constraints. The following two load cases are considered:

1. The structure is loaded in the x-direction (0° fiber orientation). Global buckling analysis is performed and system responses to be analyzed are:
   - failure criterion value $\mathbf{F}_{\text{Tsai}-\text{Wu}}$ of each lamina,
   - buckling load factor $\boldsymbol{\lambda}_{\text{cr}}$.

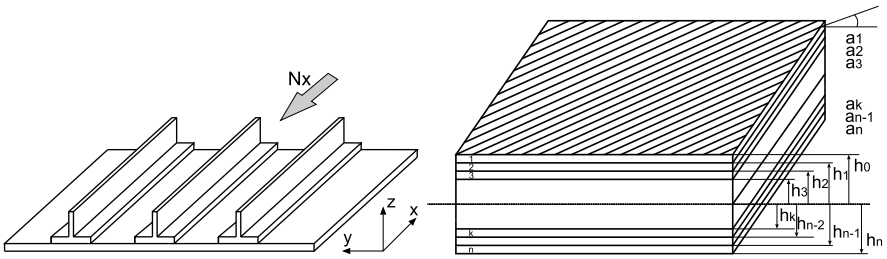


Fig. 7 Stiffened aircraft wing skin with fiber reinforced laminates

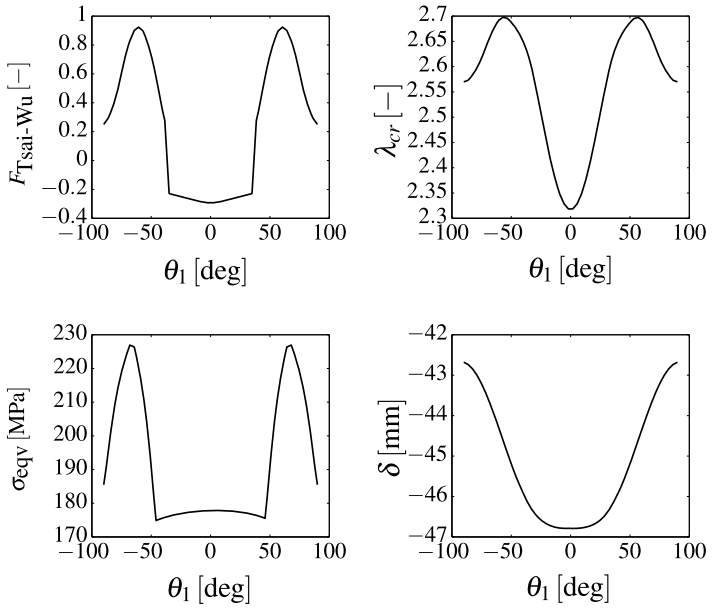


Fig. 8 One-dimensional parameter study of the fiber angle design problem. The graphs show system responses subject to $\boldsymbol{\theta}_1$

2. The structure is loaded with distributed pressure on the skin surface. The system responses to be analyzed under this load are:
   - maximum von Mises equivalent stress $\boldsymbol{\sigma}_{\text{eqv}}$,
   - tip deflection $\boldsymbol{\delta}$ under bending.

Parameter studies of the structural system are performed to explain the difficulties of approximation with traditional surrogate modeling methods. Among the system responses the following are shown as examples: $\mathbf{F}_{\text{Tsai}-\text{Wu}}$ at the first lamina of the skin, $\boldsymbol{\lambda}_{\text{cr}}$, $\boldsymbol{\sigma}_{\text{eqv}}$, and $\boldsymbol{\delta}$. First, a one-dimensional parameter study is shown, in which only the fiber angle of the first lamina of the skin $\boldsymbol{\theta}_1$ is varied (see Fig. 8). A two-

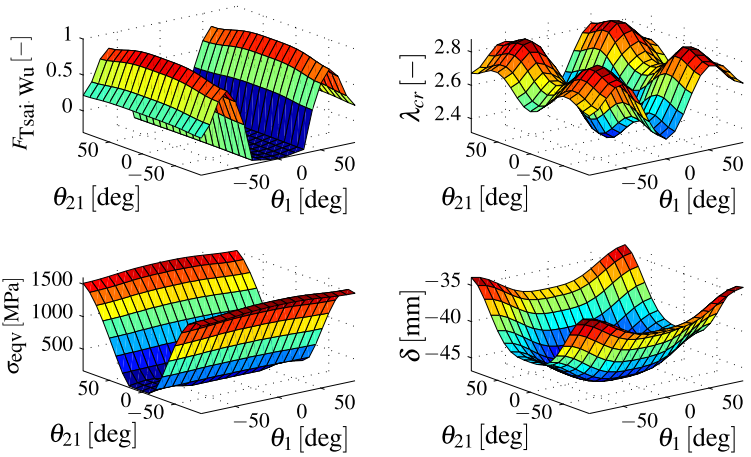


Fig. 9 Two-dimensional parameter study of the fiber angle design problem. The graphs show system responses subject to $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_{21}$

dimensional parameter study is then performed, in which only the fiber angle of the first lamina of the skin $\boldsymbol{\theta}_1$ and of the webs $\boldsymbol{\theta}_{21}$ are variable (see Fig. 9).

It is shown in the parameter studies that the system responses are highly nonlinear with respect to the design variables. A large number of sample points are required using traditional surrogate modeling methods. For example, a kriging model with quadratic regression function needs at least 1,326 samples, not to mention the oversampling factor required for adjusting the regression error and correlation parameters. It can be seen from the parameter studies that the relationships are almost incapable of being well approximated by quadratic regression functions. Higher-order polynomials are also undesirable, since the number of sample points needed will be extremely large and the computational cost extremely high.

As discussed in Sect. 4.3, the dimensionality of surrogate modeling space can be reduced by adopting elements of the stiffness matrices $\mathbf{A}^*$, $\mathbf{B}^*$, $\mathbf{D}^*$, and $\bar{\mathbf{Q}}$. In this example, laminates in all plates are symmetrically stacked so that the $\mathbf{B}^*$ matrices are zero and can be neglected. Three different sections of the structure are chosen to generate mechanical properties of the system. They are described as follows:

1. Skin: sections containing only the skin, which have 40 plies. For these sections, 12 independent matrix elements in $\mathbf{A}^*$ and $\mathbf{D}^*$ (6 in each matrix) and 6 independent matrix elements of $\bar{\mathbf{Q}}$ for each ply are to be considered.
2. Webs: sections containing only the webs, which have 60 plies. The number of matrix elements to be considered is the same as in the skin.
3. Skin + flanges: sections containing the skin and the flanges, which have 80 plies (40 plies in skin and 40 plies in flange). For these sections, the $\bar{\mathbf{Q}}$ matrix of each ply has the same element as the sections containing only the skin. As a result, only another 12 terms in $\mathbf{A}^*$ and $\mathbf{D}^*$ are to be considered.

The mechanical properties generated for the three sections are used as knowledge-based terms in surrogate modeling. For a particular system response, not all terms are necessarily included in the regression function. The relevant terms can be selected based on knowledge of the system. For example, if $\mathbf{F}_{\mathrm{Tsai-Wu}}$ of a particular lamina is considered, then knowledge-based terms to be selected are only the elements of $\mathbf{A}^*$, $\mathbf{D}^*$ of the relevant sections and $\bar{\mathbf{Q}}$ of the lamina. The selected knowledge-based terms are then used as $\mathbf{p}(\mathbf{x})$ in kriging to form the corresponding regression functions $Y(\mathbf{p}(\mathbf{x})) = \sum_{i=0}^{n} \beta_i f_i(\mathbf{p}(\mathbf{x}))$. The knowledge-based regression models for the system responses are formulated as follows.

- $\mathbf{F}_{\mathrm{Tsai-Wu}}$: the relevant section is the skin. For the first lamina of the skin, the relevant terms are elements of $\mathbf{A}^*$, $\mathbf{D}^*$ of the skin, and $\bar{\mathbf{Q}}$ of the lamina. The regression function in kriging is formulated as

$$\begin{aligned} Y_1\big(\mathbf{p}_1(\mathbf{x})\big) = \alpha_0 &+ \sum_i \sum_j \big(\beta_{A_{ij}} A^*_{ij} \bar{Q}_{1ij} + \beta_{D_{ij}} D^*_{ij} \bar{Q}_{1ij}\big) \\ &+ \sum_i \sum_j \big(\gamma_{A_{ij}} {A^*_{ij}}^2 \bar{Q}^2_{1ij} + \gamma_{D_{ij}} {D^*_{ij}}^2 \bar{Q}^2_{1ij}\big), \end{aligned} \tag{14}$$

  where 91 coefficients are to be determined.
- $\boldsymbol{\lambda}_{\mathrm{cr}}$: relevant sections are skin, webs and skin+flanges. Considering the elements of $\mathbf{A}^*$ and $\mathbf{D}^*$ of all sections, the regression function in kriging is formulated as

$$\begin{aligned} Y_2\big(\mathbf{p}_2(\mathbf{x})\big) = \alpha_0 &+ \sum_i \sum_j \big(\beta_{A_{ij}} A^*_{ij} + \beta_{D_{ij}} D^*_{ij}\big) \\ &+ \sum_i \sum_j \big(\gamma_{A_{ij}} {A^*_{ij}}^2 + \gamma_{D_{ij}} {D^*_{ij}}^2\big), \end{aligned} \tag{15}$$

  where 73 coefficients are to be determined.
- $\boldsymbol{\sigma}_{\mathrm{eqv}}$: relevant sections are webs. The relevant terms are elements of $\mathbf{A}^*$, $\mathbf{D}^*$ of the webs, and $\bar{\mathbf{Q}}$ of the first lamina of the webs. The regression function in kriging is formulated as

$$\begin{aligned} Y_3\big(\mathbf{p}_3(\mathbf{x})\big) = \alpha_0 &+ \sum_i \sum_j \big(\beta_{A_{ij}} A^*_{ij} \bar{Q}_{1ij} + \beta_{D_{ij}} D^*_{ij} \bar{Q}_{1ij}\big) \\ &+ \sum_i \sum_j \big(\gamma_{A_{ij}} {A^*_{ij}}^2 \bar{Q}^2_{1ij} + \gamma_{D_{ij}} {D^*_{ij}}^2 \bar{Q}^2_{1ij}\big), \end{aligned} \tag{16}$$

  where 97 coefficients are to be determined.
- $\boldsymbol{\delta}$: relevant sections are skin, webs, and skin+flanges. Elements of $\mathbf{A}^*$ and $\mathbf{D}^*$ of all sections are considered. The regression function in kriging is formulated as

$$\begin{aligned} Y_4\big(\mathbf{p}_4(\mathbf{x})\big) = \alpha_0 &+ \sum_i \sum_j \big(\beta_{A_{ij}} A^*_{ij} + \beta_{D_{ij}} D^*_{ij}\big) \\ &+ \sum_i \sum_j \big(\gamma_{A_{ij}} {A^*_{ij}}^2 + \gamma_{D_{ij}} {D^*_{ij}}^2\big), \end{aligned} \tag{17}$$

  where 73 coefficients are to be determined.

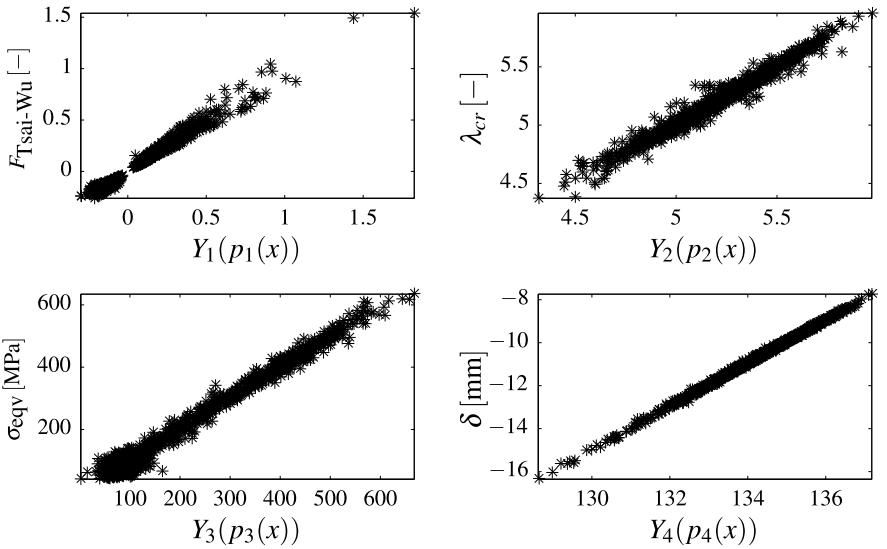


Fig. 10 Parameter study of the system responses $\mathbf{F}_{\text{Tsai-Wu}}$, $\boldsymbol{\lambda}_{\text{cr}}$, $\boldsymbol{\sigma}_{\text{eqv}}$, and $\boldsymbol{\delta}$ with respect to the knowledge-based regression functions $Y(\mathbf{p}(\mathbf{x}))$

Table 4 Results of the traditional and knowledge-based surrogate modeling methods for the fiber design problem

	n_{samp}	$\varepsilon_{\text{rrms}}(\mathbf{F}_{\text{Tsai-Wu}})$	$\varepsilon_{\text{rrms}}(\lambda_{\text{cr}})$	$\varepsilon_{\text{rrms}}(\sigma_{\text{eqv}})$	$\varepsilon_{\text{rrms}}(\delta)$
Traditional surrogate model	1,500	13.19 %	12.36 %	24.26 %	7.49 %
Knowledge-based surrogate model	300	2.81 %	0.79 %	2.41 %	0.38 %

A parameter study of the system responses with respect to the knowledge-based terms is performed (see Fig. 10). It is shown that the system responses can be well approximated by linear or quadratic combinations of the knowledge-based terms. Therefore, the dimensionality and the nonlinearity of the system are greatly reduced, and we can approximate it with much higher accuracy and efficiency.

A comparison of the performance of the traditional and the knowledge-based surrogate modeling methods is shown in Table 4, where the advantages of using the knowledge-based surrogate modeling method can be seen.

The traditional surrogate model with 1,500 samples is constructed using kriging, which contains a quadratic regression model. The knowledge-based surrogate model is constructed with 300 samples using knowledge-extended kriging (see Sect. 3.1), which contains the regression models formulated with the knowledge-based terms. As can be seen, a large approximation error results with the use of traditional surrogate modeling methods, which proves that, for high-dimensional, highly nonlinear responses, the traditional surrogate modeling method becomes incapable. On the contrary, knowledge-based surrogate models, which adopt terms representing the

mechanical properties, are able to capture the system behavior with much fewer sample points. By adopting engineering knowledge-based terms, the design space for surrogate modeling is reduced. The complexity of the relationships between the inputs and outputs of surrogate models is also remarkably reduced. As a consequence, the approximation quality is significantly improved, and the surrogate modeling process is much more efficient than a black-box surrogate modeling on a high-dimensional space.

5 Practical Example: Knowledge-Based Surrogate Modeling in Aircraft Wingbox Design Optimization

In this section an application example of an aircraft wingbox design optimization is described. The original optimization problem requires a large number of evaluations involving time-consuming FEA and CFD; therefore, surrogate-based design optimization is preferred. In order to represent the system behavior with lower computational cost, engineering knowledge is emphasized in the surrogate modeling. First, knowledge-based terms, which represent the structural mechanical properties are generated algebraically and act as a bridge between the design variables and system responses. Second, an infill criterion based on expected improvement and probability of feasibility [8] is used to verify and refine the surrogate models iteratively.

5.1 Introduction of the Aircraft Wingbox Design Optimization

The wingbox of an aircraft wing [3] to be optimized is shown in Fig. 11. A parameterized wingbox model is constructed with the ANSYS Parametric Design Language (APDL). The model is composed of shell and beam elements to represent the skins, ribs, spars, and stringers of the wingbox, as well as mass point elements to represent the engine and fuel. A cross section of the wingbox at a specific y-coordinate is represented in Fig. 12.

The structural design and analysis of an aircraft wing involves the consideration of inertial, elastic, and aerodynamic forces [13]. A thickness distribution of the aircraft wingbox is performed in order to achieve the minimal mass m, while satisfying required strength, stiffness, and stability constraints [14]. This design problem is shown mathematically as

$$\begin{aligned} &\text{minimize} && m(\mathbf{x}) \\ &\text{subject to} && \boldsymbol{\sigma}(\mathbf{x}) \leq \boldsymbol{\sigma}_{\text{crit}} \\ & && \boldsymbol{\delta}(\mathbf{x}) \leq \boldsymbol{\delta}_{\text{crit}} \\ & && -\mathbf{f}(\mathbf{x}) \leq -\mathbf{f}_{\text{load}} \\ &\text{and} && \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u . \end{aligned} \tag{18}$$

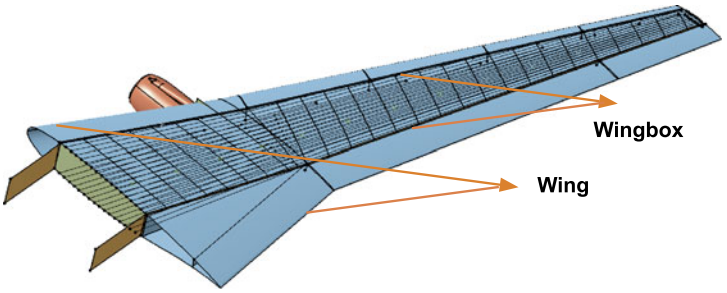


Fig. 11 CAD model of an aircraft wing, the wingbox structure (only the parts between the front and rear spars) of which is to be optimized

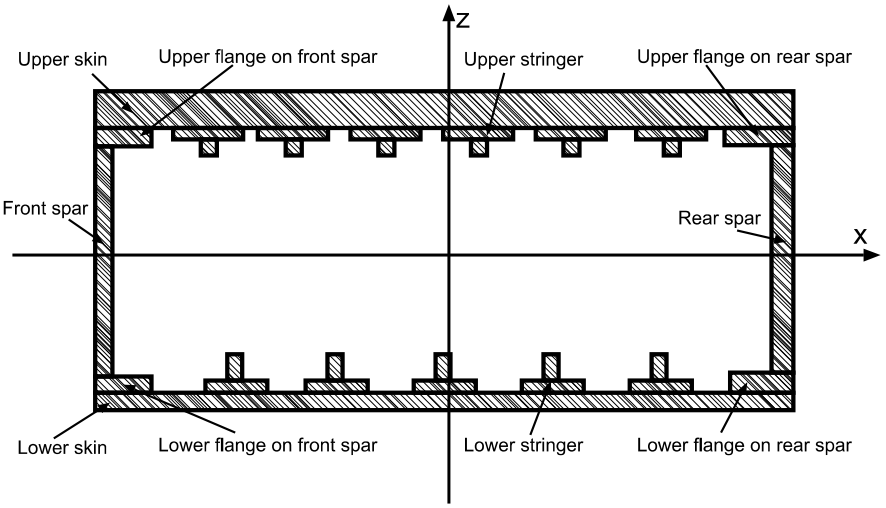


Fig. 12 Cross section of the wingbox at a specific y-coordinate

The wingbox model contains 344 design variables $\mathbf{x}$: the thicknesses of the skins, spars, ribs and stringers, which are variable spanwise and chordwise. The system responses include stresses $\boldsymbol{\sigma}$, displacements $\boldsymbol{\delta}$, and resonance frequencies $\mathbf{f}$ of the wingbox structure, which are obtained by performing FEA. A CFD simulation is used to generate an updated pressure distribution on the deformed wing. Examples of the deformed wingbox and corresponding pressure distribution are shown in Fig. 13. The pressure on the wing surface is then interpolated onto the upper and lower skin surfaces of the wingbox, as shown in Fig. 14. With the interpolation, the node displacement and pressure distribution data can be transferred between the structural model and the aerodynamic model. Structural FEA and CFD simulations are coupled to obtain the required system responses.

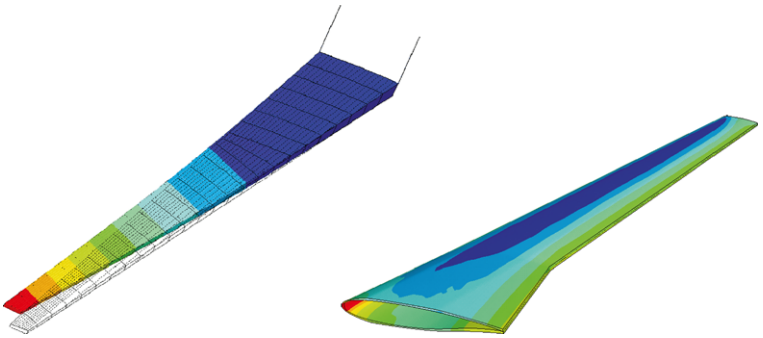


Fig. 13 Deformed wingbox (*left*) and pressure distribution on the wing skin surface (*right*)

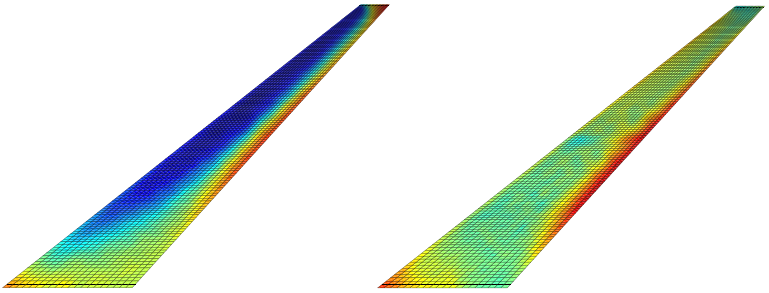


Fig. 14 Pressure distribution interpolated to the upper and lower skin surfaces of the wingbox

5.2 Knowledge-Based Surrogate Modeling Assisted Aircraft Wingbox Design Optimization

Coupled fluid-structure simulations, e.g., for the above-mentioned wing, are rather computationally expensive; therefore, surrogate models are constructed and used in the optimization process. The surrogate-based optimization process is shown in Fig. 15.

Engineering knowledge can be used to choose proper inputs and outputs for surrogate modeling. First, instead of stresses, section forces are approximated. Second, instead of the design variables, geometrical properties of the spanwise cross sections of the wingbox are generated. The geometrical properties are used to formulate knowledge-based terms for surrogate modeling. It is found that using all 26 spanwise sections can generate ill-conditioned regression functions, because the knowledge-based terms of one section are too close to those of neighboring sections. The study of conditional values in knowledge-based surrogate modeling suggests that neighbor sections should be grouped and share the same design variables. The wingbox model is grouped, spanwise, into five segments, and the total number of design variables reduces from 344 to 74. Knowledge-based terms are generated for the five segments as follows:

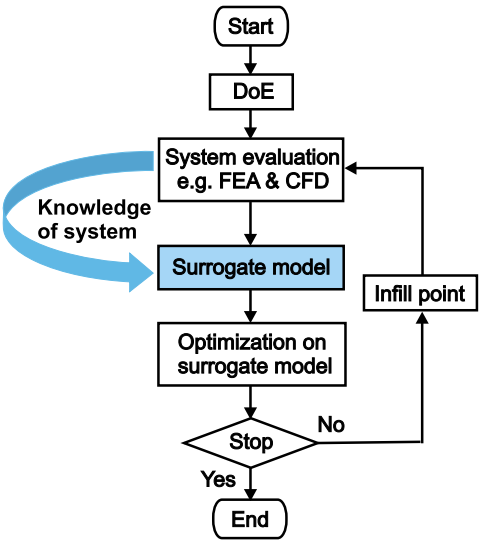


Fig. 15 Flowchart of surrogate-based wingbox design optimization

- geometrical properties for five cross sections, A_i, $I_{xx,i}$, $I_{zz,i}$, $J_{yy,i}$, $i = 1, 2, \ldots, 5$,
- thicknesses of five ribs, t_i, $i = 1, 2, \ldots, 5$.

As a result, there are only 25 terms to form a modeling space. 360 samples are used in constructing initial surrogate models. The design optimization is then carried out parallelized on the initial surrogate models from 20 different initial designs. With the infill criterion, 20 infill points are generated to refine the surrogate models. After two iterations of surrogate-based optimization with knowledge of the system, an optimum is obtained. The surrogate-based optimization results of some design variables are shown in Fig. 16. In the left graph of Fig. 16, the optimized thickness of the front spar, the rear spar, and the ribs of the five spanwise segments are shown. In these components the middle segments have the largest thickness, and the inboard segments are thicker than the outboard segments. This is done because the middle segments are susceptible to local buckling, and the inboard segments have to withstand greater stress than the outboard segments. The ribs are under shear stress and can be thinner, as the results suggest. Usually the ribs are designed with lightening holes. The right top graph of Fig. 16 shows the optimized thickness of the upper and the lower skin in the spanwise segments. The thickness generally decreases from the root to the tip, as the root segments must sustain higher stresses. The upper skin needs to be thicker because it is usually under compression, while the lower skin is under extension. The thickness of the flanges of the stringers is shown on the right bottom graph of Fig. 16, where the thickness values are smaller compared with other components. No clear trend along the chordwise direction is shown, which suggests that the chordwise influence is smaller than the spanwise influence.

The results of the adaptive surrogate modeling based optimization are shown in Table 5. The initial model refers to the initial surrogate models constructed by

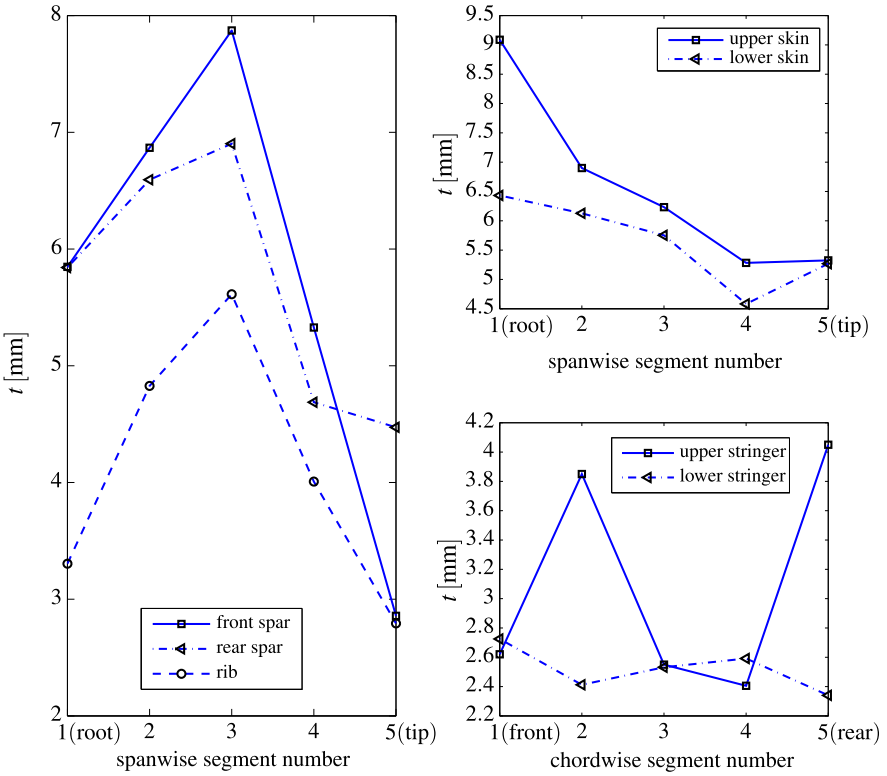


Fig. 16 Surrogate-based optimization results of the wingbox design optimization. The graphs show the thickness values of the wingbox components in spanwise segments or in the chordwise direction

Table 5 Results of the wingbox sizing optimization assisted by knowledge-based surrogate models

	n_{samp}	$\max(\varepsilon_{\text{rrms}})$	n_{feas}	m_{opt}	$g_{\max}$
Initial model	360	3.25 %	0	–	–
1st iteration model	20 infill points	2.08 %	2	1318.5	−0.0302
2nd iteration model	20 infill points	1.99 %	15	1302.6	−0.0172

knowledge-based surrogate modeling. The first and second iteration models refer to the knowledge-based surrogate models refined by additional sample points from the infill criterion. $\max(\varepsilon_{\text{rrms}})$ is the maximum relative root mean square error of the approximation, tested on another set of points generalized in parallel with the sample points. n_{feas} is the number of feasible designs obtained from 20 parallelized optimization processes. $g_{\max}$ is the maximum constraint violation value, for which a negative value indicates a feasible design. It can be seen that the initial knowledge-

based surrogate models have good accuracy, which is then improved in two iterations with infill points. Note that no feasible design can be guaranteed by optimizing on the initial surrogate models, and the feasibility of surrogate-based optimization results should always be verified. When more information on the design problem is obtained, the knowledge-based surrogate models are refined and are finally suitable to be used as system equations in the optimization process.

5.3 Discussion of Computational Effort

The aircraft wingbox design optimization problem contains 344 design variables; therefore, thousands of system evaluations are required using either numerical gradient-based optimization algorithms or evolutionary algorithms. As already described in Sect. 5.1, high-fidelity system evaluation of this problem is computationally expensive and time consuming, which prevents efficient optimization. If no surrogate models are established, for design optimization of this problem, especially in the case of finite differencing, the total computational effort would be specifically large. As a result, the surrogate modeling technique is required. However, a large number of system evaluations are still necessary using conventional surrogate modeling methods. For example, the full second-order polynomial regression would need at least 59,685 sample points, resulting in a massive computational effort. Using the knowledge-based surrogate modeling technique, 25 system geometrical properties are used as knowledge-based terms in kriging to assist the design optimization process. An optimum design is obtained with altogether 400 high-fidelity system evaluations for the provision of sample points. The design optimization is performed with the surrogate models, which requires a nearly negligible computational effort compared with the high-fidelity system evaluations.

6 Conclusions

In a large and complex engineering design optimization, the system equations are usually expensive to evaluate, and gradient-based optimization might require a large number of system evaluations. As a result, surrogate models are used to facilitate engineering design optimization and reduce the computational effort. Aerospace structures are often composed of extruded aluminum and various composite materials, which are modeled as beam and shell elements in engineering analysis. It is suggested that a knowledge of these materials and elements as well as the structural system be adopted in surrogate modeling. With engineering knowledge, terms which are capable of representing the system properties, such as stiffness, strength, and stability, can be generated and used in surrogate models. Thereby, the number of sample points for surrogate modeling can be reduced, and a higher approximation efficiency and accuracy can be obtained. Compared with traditional surrogate modeling methods, knowledge-based surrogate modeling is more appropriate and reliable to assist in engineering design optimization.

Acknowledgements This work is supported partially by the China Scholarship Council (CSC) and is also based in part on investigations of the Collaborative Research Center (SFB) Transregio 10 project C1, which is supported by the German Research Foundation (DFG).

References

1. Agarwal, B.D., Broutman, L.J.: Analysis and Performance of Fiber Composites. Wiley, New York (1980)
2. Alvarez, L.F., Toropov, V.V., Hughes, D.C., Ashour, A.F.: Approximation model building using programming methodology: applications. In: Second ISSMO, AIAA Internet Conference on Approximations and Fast Reanalysis in Engineering Optimization (2000)
3. Armanini, S.: Construction of a finite element reference model of a wing. Semester thesis, Institute of Lightweight Structures, Technical University of Munich, Germany (2011)
4. Baier, H., Seesselberg, C., Specht, B.: Optimierung in der Strukturmechanik. LSS Verlag, Dortmund (2006)
5. Bandler, J.W., Cheng, Q.S., Dakroury, S.A., Bakr, M.H.: Space mapping: the state of the art. IEEE Trans. Microw. Theory Tech. **52**(1), 337–361 (2001)
6. Baudat, G., Anouar, F.: Kernel-based methods and function approximation. In: International Joint Conference on Neural Networks, vol. 2, pp. 1244–1249 (2001)
7. Clarke, S.M., Griebsch, J.H., Simpson, T.W.: Analysis of support vector regression for complex engineering analyses. J. Mech. Des. **127**, 1077–1087 (2005)
8. Forrester, A., Sóbester, A., Keane, A.: Engineering Design via Surrogate Modelling. Wiley, Chichester (2008)
9. Giunta, A.A., Eldred, M.S.: Implementation of a trust region model management strategy in the DAKOTA optimization toolkit. In: 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA (2000)
10. Gupta, K.K., Meek, J.L.: Finite Element Multidisciplinary Analysis. AIAA, Reston (2003)
11. Hastie, T., Tibshirani, R., Friedman, J.: Data Mining, Inference, and Prediction. Springer, New York (2008)
12. Inman, D.J.: Engineering Vibration. Prentice Hall International, Upper Saddle River (2001)
13. Hodges, D.H., Poerce, G.A.: Introduction to Structural Dynamics and Aeroelasticity. Cambridge Aerospace Series. Cambridge University Press, Cambridge (2011)
14. Niu, M.C.Y.: Airframe Structural Design. Conmilit Press, North Point (1988)
15. Queipo, N.V., Haftka, R.T., Shyy, W., Goel, T., Vaidyanathan, R., Tucker, P.K.: Surrogate-based analysis and optimization. Prog. Aerosp. Sci. **41**, 1–28 (2005)
16. Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design analysis of computer experiments. Stat. Sci. **4**(4), 409–423 (1989)
17. Shan, S., Wang, G.G.: Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. Struct. Multidiscip. Optim. **41**, 219–241 (2010)
18. Sieberz, K., Bebber, D., Hochkirchen, T.: Statistische Versuchsplanung. Springer, Berlin (2010)
19. Simpson, T.W., Lin, D.K.J., Chen, W.: Sampling strategies for computer experiments: design and analysis. Int. J. Reliab. Appl. **2**(3), 209–240 (2001)
20. Todoroki, A., Sekishiro, M.: Stacking sequence optimization to maximize the buckling load of blade-stiffened panels with strength constraints using the iterative fractal branch and bound method. Composites, Part B **39**(5), 842–850 (2008)
21. Wang, H., Li, G.: Min-median-max metamodel-based unconstrained nonlinear optimization problems. Struct. Multidiscip. Optim. **45**(3), 401–415 (2012)

Switching Response Surface Models for Structural Health Monitoring of Bridges

Keith Worden, Elizabeth J. Cross, and James M.W. Brownjohn

Abstract Structural health monitoring (SHM) is the discipline of diagnosing damage and estimating safe remaining life for structures and systems. Often SHM is accomplished by detecting changes in measured quantities from the structure of interest; if there are no competing explanations for the changes, one infers that they are the result of damage. If the structure of interest is subject to changes in its environmental or operational conditions, one must understand the effects of these changes in order that one does not falsely claim that damage has occurred when one observes measurement changes. This problem—the problem of *confounding influences*—is particularly pressing for civil infrastructure where the given structure is usually openly exposed to the weather and may be subject to strongly varying operational conditions. One approach to understanding confounding influences is to construct a data-based *response surface model* that can represent measurement variations as a function of environmental and operational variables. The models can then be used to remove environmental and operational variations so that change detection algorithms signal the occurrence of damage alone. The current chapter is concerned with such response surface models in the case of SHM of bridges. In particular, classes of response surface models that can *switch* discontinuously between regimes are discussed.

Keywords Response surface models · Treed Gaussian processes · Structural health monitoring · Environmental and operational variations · Bridge case studies

K. Worden (✉) · E.J. Cross
Dynamics Research Group, Department of Mechanical Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, UK
e-mail: k.worden@sheffield.ac.uk

E.J. Cross
e-mail: e.j.cross@sheffield.ac.uk

J.M.W. Brownjohn
Vibration Engineering Section, Department of Civil and Structural Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, UK

S. Koziel, L. Leifsson (eds.), *Surrogate-Based Modeling and Optimization*,
DOI 10.1007/978-1-4614-7551-4_14,

1 Introduction

Very briefly, structural health monitoring (SHM) is the engineering discipline concerned with inferring the state of health of a structure or system from measurements obtained from sensors permanently installed on the structure or within the system [1]. It is possible to exploit a very diverse range of sensor technologies in the implementation of an SHM system, but one of the more common choices is to monitor dynamical response using accelerometers, etc. This choice leads to vibration-based SHM, and this is the main choice considered in this chapter.

It is critical to note that an SHM system is much more than a sensor network. It is almost always the case that the information about the health of the structure is well hidden in the raw time series data acquired by sensing. The reason is that small incipient damage will not usually cause a major departure from the dynamical behaviour of the healthy structure. Because of this fact, the vital ingredient in any SHM system is an inference engine which constructs low-dimensional data vectors called *features* in which the effect of damage is much more visible. An example of a damage-sensitive feature vector often used in vibration-based SHM would be a set of the natural frequencies or resonance frequencies of the structure of interest. Natural and resonance frequencies are functions of the structural stiffness and will (usually) decrease when damage—such as a fatigue crack—causes a local reduction in stiffness. Determining natural frequencies from the raw time data is one example of *feature extraction* as it is referred to in the context of pattern recognition or machine learning [1]. Once damage-sensitive features have been determined, the SHM inference engine can proceed to an analysis which provides diagnostic and prognostic information about the health of the structure.

One of the major problems associated with SHM is that features may change as a result of mechanisms other than damage and one does not usually wish to raise an alarm as a result of these benign changes. These other influences on the features will be referred to here as *confounding influences*; they most often arise in the context of engineering as the result of changes in the environment or operating conditions of the structure of interest. For the bridges discussed in this chapter, ambient temperature is an environmental variable which strongly affects the SHM features, while traffic loading is an equally important operational influence. If natural frequencies are to be used as features for SHM, it has long been known that variations in the frequencies due to temperature changes can mask variations due to damage [2]. In order to implement damage detection by detecting changes in features one must clearly produce features that are sensitive to damage but insensitive to environmental and operational variations, or alternatively, one must project out from the features the influence of the benign variations. This process is commonly referred to in the SHM literature as *data normalisation*; various techniques can be applied and a good, fairly recent, survey of the field can be found in [3].

Among the techniques available for data normalisation, one of the simplest is a regression-based approach. This relies on the availability of measurements of the environmental or operational variables of interest. When the features for SHM are based on the dynamics of the structure—as in vibration-based SHM—the response variables almost always change on a much shorter time scale than the confounding

influences. For example, accelerations measured on a bridge will have frequencies associated with tens of hertz, while cycles of variation associated with temperature or traffic will be on scales of hours or more. This means that time histories acquired over hours or days will show their main variation as a result of the confounding influences with the dynamical behaviour superimposed as a form of high-frequency 'noise'. Fitting a regression model to such data with the environmental or operational variables as the independent variables will then capture only the dependence on the confounding influences, and predictions from this model can then be subtracted from subsequent data, with the remaining residual (hopefully) sensitive *only* to damage. Regression models used in this context are often called *response surface models* and can vary in sophistication from simple polynomials [4] to state-of-the-art structures derived from modern machine learning theory like artificial neural networks and support vector machines [5, 6]; examples from both ends of the spectrum will be presented in this paper. Complications can arise if the confounding influences cause discontinuous changes in the features as the ambient variables change; for example if polynomial models are selected, discontinuous behaviour may force the choice of very high-order polynomials with the result that very many coefficients need to be estimated. If the response surface models have the capability to *switch* between simple (e.g. linear) submodels, the number of parameters for estimation from the data can be much smaller; such models are often referred to as *parsimonious*. Parsimonious models are always selected where possible as they require less training data for their estimation problem, and data in engineering problems can often be in short supply.

This chapter will illustrate the use of switching models in the context of the SHM of bridges. Illustration will be via two case studies of real bridges. In the first case, relating to the Tamar Bridge, it will be shown how engineering insight allowed a switching model based on simple polynomials to be hand-crafted; in the second case, relating to the Z24 Bridge, a class of powerful switching models that require no *a priori* engineering insight will be demonstrated.

2 Case Study 1: The Tamar Bridge

2.1 Background

The Tamar Bridge (Fig. 1) in the south-west of the UK carries a major road across the River Tamar from the town of Saltash in the county of Cornwall to the city of Plymouth in Devon. The bridge was originally constructed to a classical suspension bridge design and, for some time, was the longest in the UK. On its construction in 1961, the bridge had a main span of 335 metres and side spans of 114 metres; including anchorage and approach, the overall length of the structure was 643 metres. The bridge has two reinforced concrete towers with a height of 73 metres with the bridge deck suspended at mid-height. The original deck had three lanes and was of a composite construction (concrete, asphalt and longitudinal beams).

Fig. 1 The Tamar suspension bridge

In the late 1990s it was found that the bridge would not be able to meet a new EU directive demanding that bridges should carry lorries up to 40 tonnes in weight. Since restricting use by such vehicles would damage the local economy, the bridge was strengthened and widened. While the main deck was being replaced, two temporary relief lanes were added; however it was finally decided to make the extra lanes part of the permanent structure. The major upgrade gave rise to interest in the bridge performance, and various sensor systems were installed to measure parameters such as tensions on stays added during the upgrade, wind velocity and structural temperature. Deflection and tilt sensors were also added.

Interest in monitoring the bridge was maintained and currently three systems are in place and running. For the purposes of this chapter, the relevant system is a vibration-based system that was installed by members of the Vibration Engineering Section of the Department of Civil and Structural Engineering in the University of Sheffield in 2006. The system is based around a set of accelerometer sensors installed to monitor dynamic behaviour of the deck and some selected cables. Eight accelerometers are assigned in orthogonal pairs to four stay cables and three sensors measure deck accelerations. The monitoring system records time data at a sampling rate of 64 Hz at 10-min intervals; this data is passed to a computer which carries out an automated modal analysis in order to extract the natural frequencies of the structure. For further details, the reader can consult [4], a reference which also discusses the other monitoring systems currently in operation on the bridge.

2.2 Analysis

In a recent study [4], simple polynomial response surface models were fitted to the natural frequency data from the Tamar Bridge in order to gain insight into which environmental and operational effects were driving the feature variation. The analysis

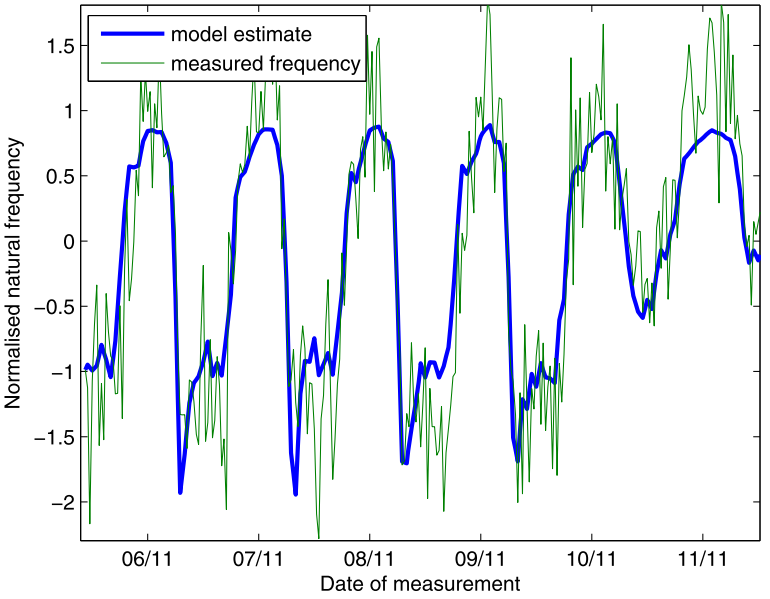


Fig. 2 Linear model of first deck modal frequency with traffic loading only

yields a type of sensitivity analysis that can be used to rank variables according to their effect on natural frequencies. The response surface models used were simple multinomials; if the operational and environmental variables are grouped in a vector $\underline{\theta} = \{\theta_1, \ldots, \theta_d\}$, the models take the form

$$f_i(\underline{\theta}) = \underline{h}(\underline{\theta})^T \underline{\beta} \tag{1}$$

where f_i is the natural frequency under study, $\underline{h}(\underline{\theta})$ is a vector of multinomial basis functions and $\underline{\beta}$ their corresponding vector of coefficients. In (almost) the simplest case, one can take $\underline{h}(\underline{\theta}) = (1, \underline{\theta}^T)^T$ and the model will be linear in the θ_i. Models of the form (1) have the advantage that simple least-squares methods allow coefficient estimation. A major advantage of linear response surface models is that (if the θ_i are standardised), the coefficients in the expansion model give an indication of the importance of the expansion variables. As an illustration, the results of fitting a linear response surface to the first natural frequency of the bridge will be given here. In this case, it was clear from the coefficient estimates that the dominant effect on the natural frequency was from traffic loading; Fig. 2 shows how well the behaviour is captured using this single independent variable in the model.

A more careful analysis of the models was carried out by using F- and T-statistics in order to establish which coefficients were *statistically significant*. For the details of this analysis, the reader can refer to [4]; however, the main result was to indicate that there was a quite small but statistically significant effect of temperature which improved the response surface models slightly from those based on traffic alone. However, a careful consideration of long periods of Tamar data showed

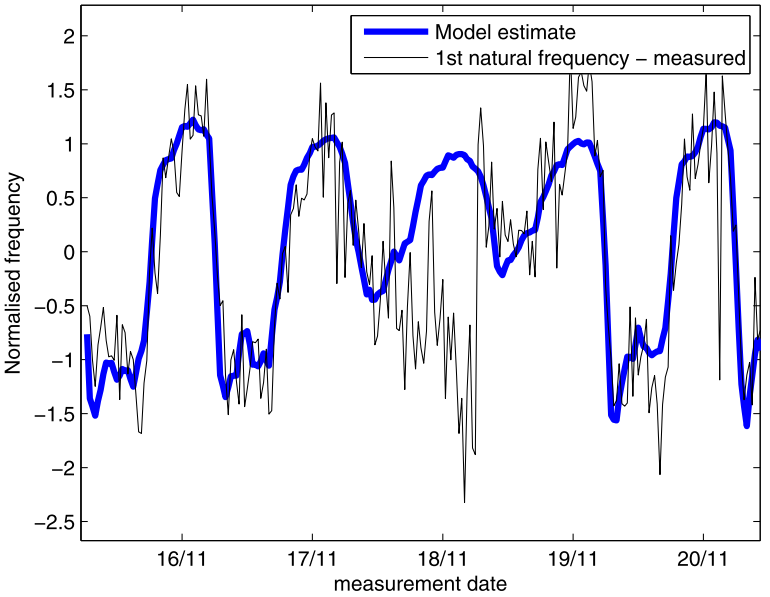


Fig. 3 Linear model of first deck modal frequency with traffic loading only; a breakdown of the model is clearly visible in the data

a small number of anomalous regions where the models completely failed to capture the behaviour regardless of whether temperature variation was added or not. An example is shown in Fig. 3.

A careful consideration of the variables showed that the times at which the linear response surface failed were associated with high wind speeds, with the wind crossing the bridge deck in the transverse direction to the span, i.e. from the north or south. Further investigation showed that two regimes were visible in the vibrational behaviour of the bridge: below 25 mph wind speed (transverse to the span direction), the vertical acceleration of the bridge showed no significant dependence on wind speed; however, above 25 mph, vertical acceleration increased linearly with wind speed (Fig. 4). For easterly or westerly winds, there was no marked dependence of the vertical acceleration on wind speed. In order to try and capture this behaviour, and its possible reflection in the natural frequency, it would have been possible to add wind speed as an input to the response surface model; however, this would have forced adoption of a nonlinear model in order to allow switching between regimes. In this specific case a simpler solution was available; adding the RMS vertical acceleration as an input variable incorporated the required switching behaviour while still allowing a linear response surface model. The addition of the additional variable on the basis of engineering insight allowed a much more accurate representation of the natural frequency in the formerly anomalous region (Fig. 5).

The advantage of the new model is very clear here: by allowing the simple linear model to still operate in the two separate regimes, the number of coefficients in the

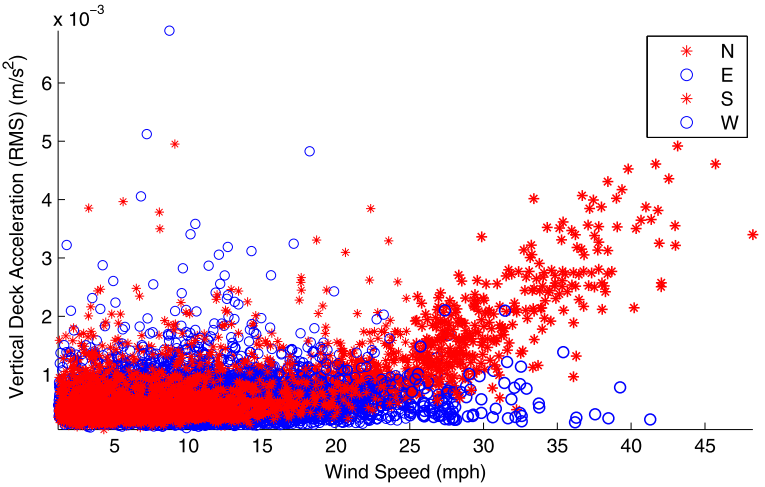


Fig. 4 RMS vertical acceleration of deck as a function of wind speed, sorted in terms of wind direction

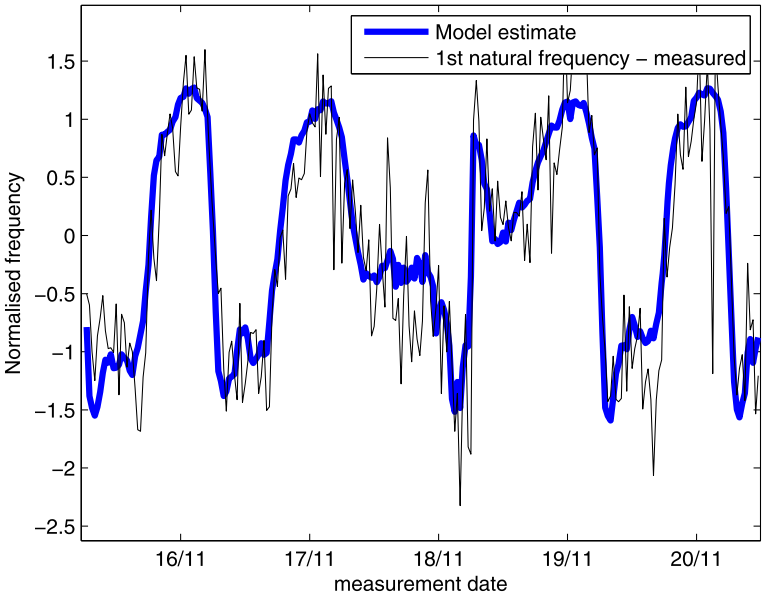


Fig. 5 Linear model of first deck modal frequency with traffic loading and RMS vertical acceleration

model is kept very small. Any attempt to capture the behaviour here in a polynomial model incorporating wind speed would have resulted in the use of a high-order polynomial and would have likely resulted in orders of magnitude greater numbers

of coefficients. One perceived problem with the approach taken here might be that a certain amount of engineering insight and expertise was necessary in order to establish the model form. Fortunately, there are response surface models available that can identify and capture switching behaviour based on data alone with minimal user intervention. The issue is that these models are much more complex to formulate in the first place. In the remainder of the paper a group of such models will be introduced and illustrated; in order to minimise the complexity of the presentation, a case study involving a single important environmental variable will be considered.

3 Switching Regression Models: Regression Trees

3.1 Regression Trees

The idea of a regression tree is fairly simple to state (much of the theory and practice of such trees can be attributed to the work of Breiman and colleagues, and a good reference is [7]). The idea is to partition the independent variable space into regions over which the response behaviour is smooth and fit low-order regression models over each region. If the partitioning is carried out by hand, the resulting problem is still amenable to linear least-squares methods. However, the idea of a regression tree in general is that the partitions are determined from the data as part of the modelling problem; this renders the estimation problem highly nonlinear and alternatives to least squares are needed. Breiman and co-workers established a greedy algorithm for fitting the trees that gave good (but suboptimal) solutions. If an effective partition of the data is found, linear regression models over each distinct region can give excellent results; however, in principle, *any* algorithm can be used once the data have been partitioned into sensible regions. Once the concept of *classification and regression trees* (CART) was established, arguably the next major advance was the development of a Bayesian framework for the algorithm [8, 9]. The new algorithm was based on rigorous concepts of probability theory and proved an effective departure from the greedy algorithm. In Bayesian CART, a prior probability distribution was proposed over all possible tree structures as well as all possible coefficients. This was then refined by using the data to determine which tree was supported by the greatest evidence. The original formulation is too complicated to describe here without taking this paper a long way from its illustrative objectives. In the original Chipman formulation, all the regression models within the tree were linear. This restriction was later removed by Gramacy [10], who replaced the linear models by more powerful *Gaussian process* (*GP*) models; Gramacy's work also extended the Bayesian formulation of the problem significantly.

3.2 Gaussian Processes

Gaussian process (GP) regression has recently become a popular technique in machine learning, although its roots go back many years [11]. (In fact, the subject had

its roots in geotechnical signal processing as early as the 1950s, where the idea was referred to as *Kriging* [12].) In essence, GPs are an extension of the multivariate Gaussian probability distribution. Unlike most forms of regression models that return a crisp value $f(\underline{x})$ for any given $\underline{x}$, a GP *returns a Gaussian probability distribution*. The GP is thus a Gaussian distribution over functions. Among the advantages of the GP for regression purposes are its principled statistical (Bayesian) foundations and the fact that it automatically returns a confidence interval for predictions. GPs adhere to the Bayesian paradigm in the sense that a number of prior assumptions are made about the function being modelled, and then training data (samples of the features) are used to update and evaluate a posterior distribution over functions. A key assumption is that the model is a smooth function of its inputs and this allows extra information concerning the response to be gained at reduced computational cost. An extended variant of the GP algorithm was developed by O' Hagan and colleagues [13] using additional ideas from Bayesian probability, and it is this variant that will be described briefly here. The Bayesian formulation makes the incorporation into a Bayesian regression tree formulation more direct. Because the implementation of the GP algorithm (unlike its derivation) is straightforward to state, it will be given here in a little detail.

For any set of n input points $\{\underline{x}_1, \ldots, \underline{x}_n\}$ (which represent the values of the identification parameters for the specific problem considered here), each of dimension d, the prior beliefs about the corresponding outputs can be represented by a multivariate normal distribution, the mean of which is a least-squares regression fit through the training data,

$$E\big[f(\underline{x})|\underline{\beta}\big] = m(\underline{x}) = \underline{h}(\underline{x})^T\,\underline{\beta} \tag{2}$$

where $\underline{h}(\underline{x})^T$ is a specified (vector) regression function of $\underline{x}$, and $\underline{\beta}$ is the corresponding vector of coefficients. For simplicity, $\underline{h}(\underline{x})^T$ was chosen here to be $(1, \underline{x}^T)$, representing a linear regression (this can be extended to higher polynomial fits if required or simply set to zero). The covariance between output points is given as

$$\operatorname{cov}\big[f(\underline{x}), f\big(\underline{x}'\big)\big|\sigma^2, B\big] = \sigma^2 c\big(\underline{x}, \underline{x}'\big) \tag{3}$$

where σ^2 is a scaling factor (sometimes called the *height parameter*) and B is a diagonal matrix of (inverse) length scales, representing the roughness of the output with respect to the individual input parameters. The covariance function commonly adopted, and used here, is a squared exponential function of the form

$$c(\underline{x}, \underline{x}') = \exp\big[-\big(\underline{x} - \underline{x}'\big)^T B\big(\underline{x} - \underline{x}'\big)\big] + \sigma_n^2 \tag{4}$$

where the *nugget* σ_n^2 is a hyperparameter accounting for measurement noise. (For simplicity, the closed-form expressions for the case $\sigma_n = 0$ are given below, although all the results presented later were obtained without this restriction.)

These equations complete the prior specification of the problem; the *posterior distribution* is then found by conditioning the prior distribution on the training data $\underline{y}$ (the vector of output points corresponding to the input training set) and integrating out (or marginalising over) the hyperparameters σ^2 and β. The calculation is

straightforward but very time-consuming, a detailed walkthrough can be found in [14]. The integrals involved are usually all Gaussian, and although the expressions are almost always very complicated, the results can be given in closed form. The result is a Student's t-process, conditional on B and the training data,

$$\left[f(\underline{x})|y, B\right] \sim t_{n-q}\left(m^*(\underline{x}), \hat{\sigma}^2 c^*(\underline{x}, \underline{x})\right) \tag{5}$$

where

$$m^*(\underline{x}) = \underline{h}(\underline{x})^T \underline{\hat{\beta}} + \underline{t}(\underline{x})^T A^{-1}(\underline{y} - H\underline{\hat{\beta}}) \tag{6}$$

$$\begin{aligned} c^*(\underline{x}, \underline{x}') = c(\underline{x}, \underline{x}') - \underline{t}(\underline{x})^T A^{-1} \underline{t}(\underline{x}') \\ + \left(\underline{h}(\underline{x}) - \underline{t}(\underline{x}) A^{-1} H\right)\left(H^T A^{-1} H\right)\left(\underline{h}(\underline{x}') - \underline{t}(\underline{x}') A^{-1} H\right)^T \end{aligned} \tag{7}$$

$$\underline{t}(\underline{x})^T = \left(c(\underline{x}, \underline{x}_1), \ldots, c(\underline{x}, \underline{x}_n)\right) \tag{8}$$

$$H^T = \left(\underline{h}(\underline{x}_1), \ldots, \underline{h}(\underline{x}_n)\right) \tag{9}$$

$$A = \begin{pmatrix} 1 & c(\underline{x}_1, \underline{x}_2) & \ldots & c(\underline{x}_1, \underline{x}_n) \\ c(\underline{x}_2, \underline{x}_1) & 1 & & \vdots \\ \vdots & & \ddots & \\ c(\underline{x}_n, \underline{x}_1) & \ldots & & 1 \end{pmatrix} \tag{10}$$

$$\underline{\hat{\beta}} = \left(H^T A^{-1} H\right)^T H^T A^{-1} \underline{y} \tag{11}$$

$$\hat{\sigma}^2 = \frac{\underline{y}^T (A^{-1} - A^{-1} H (H^T A^{-1} H) H^T A^{-1}) \underline{y}}{n - d - 3} \tag{12}$$

$$\underline{y} = \left(f(\underline{x}_1), \ldots, f(\underline{x}_n)\right)^T \tag{13}$$

Determination of this model is basically an exercise in machine learning, and therefore its quality is critically dependent on the number and distribution of training data points in the input space, and the values of the hyperparameters. The expressions for $\underline{\hat{\beta}}$ and $\hat{\sigma}^2$ shown above are the result of marginalisation; however, it can be shown that they actually coincide with least-squares estimates. The diagonal matrix of roughness parameters B cannot generally be integrated out analytically, and it is usually evaluated using maximum likelihood estimation or a Markov chain Monte Carlo (MCMC) method. This calculation typically represents the most computationally intensive part of the process.

This almost completes the basic description of the models used here: *treed Gaussian processes* (TGPs). Although the actual implementation is too complex to explain here in any detail, the basic ingredients have been covered. The TGP partitions the variable space in much the same way as a Bayesian CART and then fits GP regressors over each independent region. The software used for modelling in this work is the TGP package written by Gramacy in the language R [15]. A useful feature of

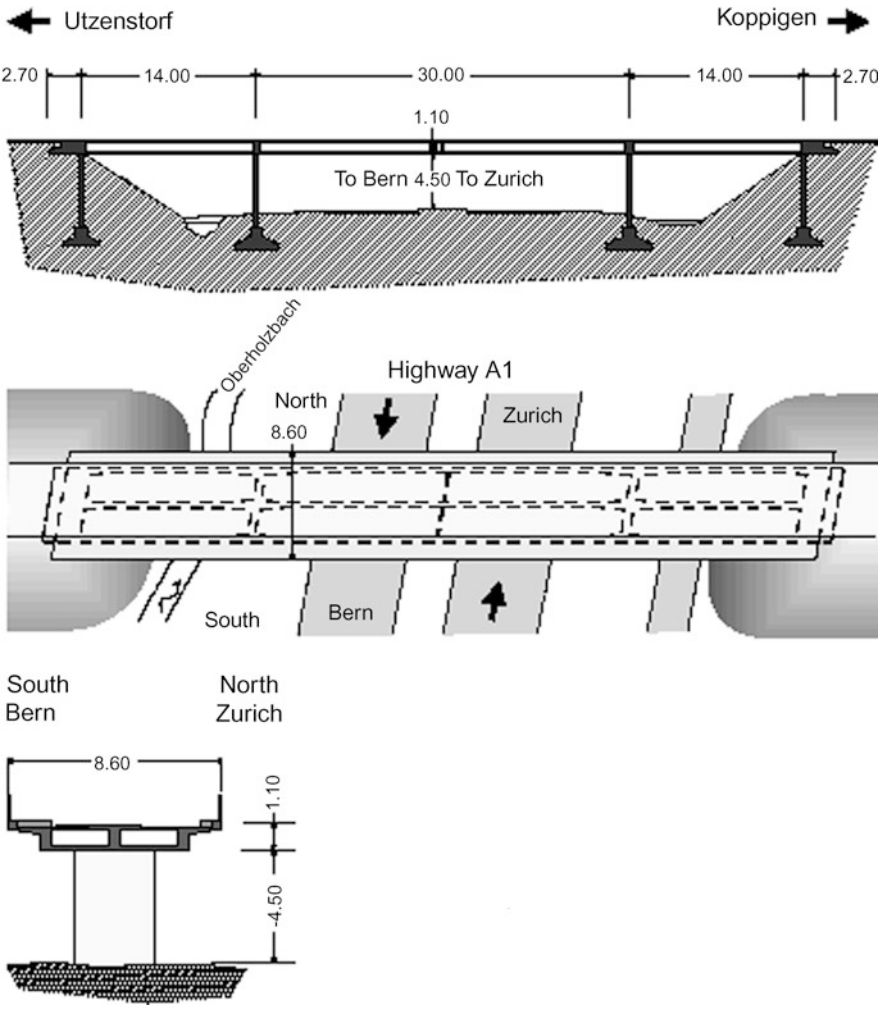


Fig. 6 Z24 Bridge longitudinal section and top view [17]

the TGP software is the *limiting linear model* [16]. Because of the Bayesian framework adopted for the TGP, the software can essentially assess if the evidence for a full GP model outweighs that for a linear model; if the software judges that a linear model is sufficient over a given region of the feature, it switches to the simpler representation.

In the next section, the use of the treed models is demonstrated on data from the Z24 Bridge, a situation in which a single environmental variable provided the confounding influences and required removal from the SHM feature data.

4 Case Study 2: The Z24 Bridge

4.1 Background

The Z24 Bridge, a pre-stressed concrete highway bridge in Switzerland (Fig. 6), was subject to a comprehensive monitoring campaign under the 'SIMCES project' [18]. Prior to its demolishment in the late 1990s, the bridge had become a landmark benchmark study in SHM. The monitoring campaign, which spanned a whole year, tracked modal parameters and included extensive measurement of the environmental factors affecting the structure, such as air temperature, soil temperature, humidity etc. The Z24 monitoring exercise was an important study in the history of SHM developments, because towards the end of the monitoring campaign researchers were able to introduce a number of realistic damage scenarios to the structure. In order, these scenarios were [19]:

- Pier settlement
- Tilt of foundation followed by settlement removal
- Concrete spalling
- Landslide
- Concrete hinge failure
- Anchor head failure
- Tendon rupture.

The papers immediately produced as a result of the COST action were [20–23], although many have followed.

The SHM features of interest here are the natural frequencies of the bridge, which were tracked over the period of a year and additionally over the period where the bridge was damaged according to the various scenarios. Modal properties of the bridge were extracted from acceleration data [24]. Figure 7 shows a time history of the four natural frequencies between 0–12 Hz of the bridge. The solid vertical line marks the start of the period where the different levels of damage (starting with pier lowering) were introduced. Gaps where the monitoring system failed have been removed. On inspection of Fig. 7, one can see that the natural frequencies of the bridge are by no means stationary. There are some large fluctuations in the first half of the time history before the introduction of any damage. These fluctuations occurred during periods of very cold temperatures and have been associated with an increase in stiffness caused by freezing of the asphalt layer on the bridge deck. The natural frequency time histories are, therefore, another good example of how damage-sensitive parameters can also be very sensitive to environmental variations—in this case temperature.

As the natural frequencies in their current form would not be suitable to monitor as a damage-sensitive feature, some action must be taken to remove the variable set's sensitivity to temperature. Although the regression approach discussed in this paper is one of the simpler methods conceptually, it is indicated here as, in the Z24 case, the modal properties of the bridge are nonlinearly dependent on temperature (as an example, Fig. 8 plots how the first natural frequency changes with temperature). The bilinear form of the dependence on temperature also means that the switching

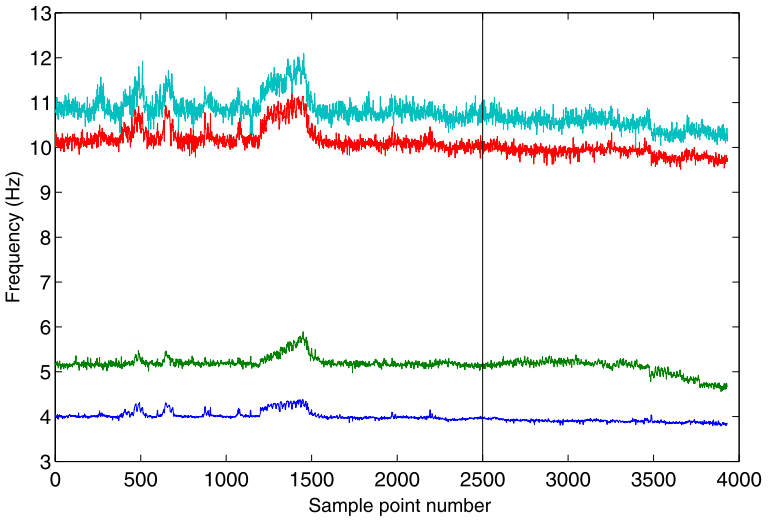


Fig. 7 Time histories of the extracted natural frequencies of the Z24 Bridge, monitored over one year including a period when damage was introduced

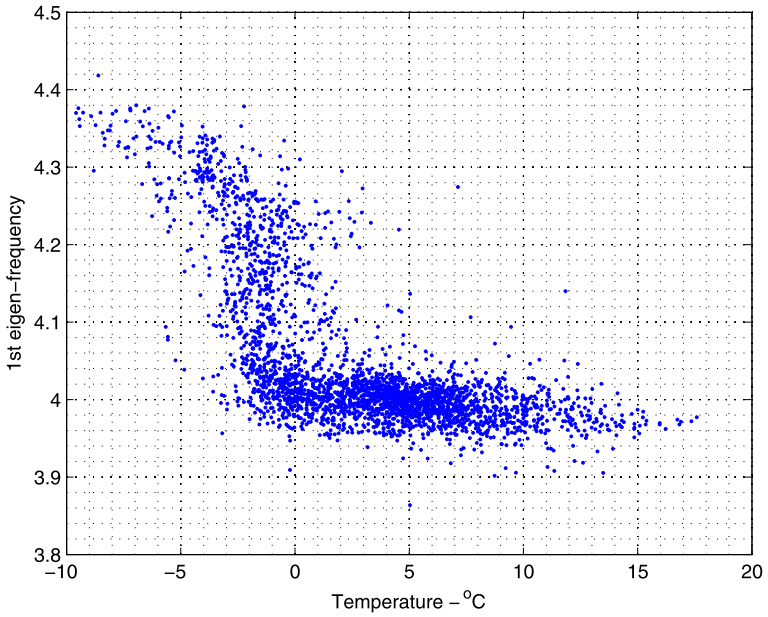


Fig. 8 First natural frequency's nonlinear dependence on temperature

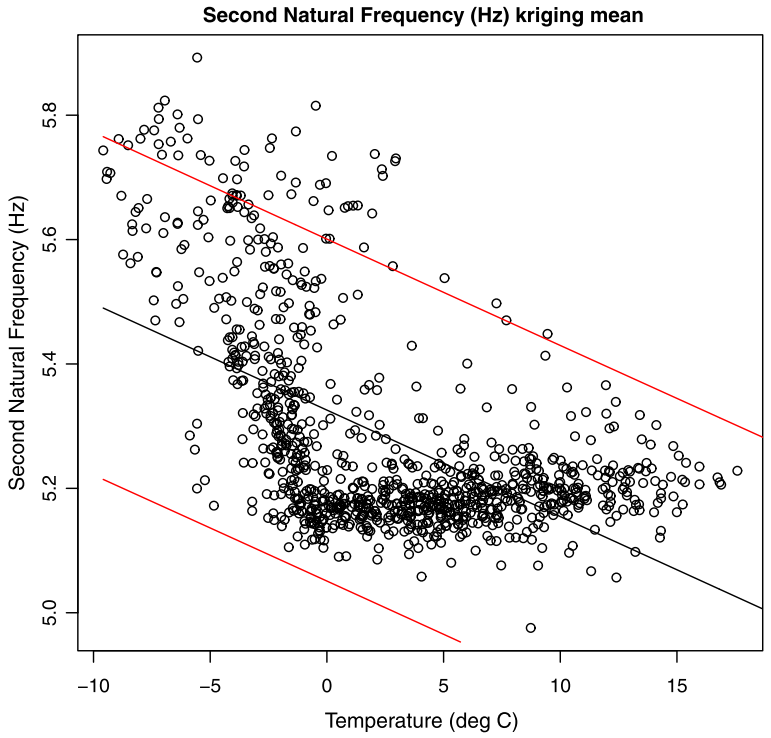


Fig. 9 Bayesian linear model of Z24 second natural frequency as a function of temperature: training data

models should prove useful. For the purposes of this case study, the second natural frequency will be discussed, as the first frequency is rather insensitive to damage.

4.2 Analysis

As a benchmark, the first analysis of the data simply fitted a linear regression model. However, the analysis was carried out in the Bayesian framework in order to provide confidence intervals for the model predictions. A rather long-winded way of accomplishing this is to train a GP and set the height parameter σ^2 to zero; however, more direct ways are easily established, as discussed in the first chapter of [11]. In order to develop the models in a principled manner, the natural frequency data were divided into a training set (to establish the model) and a testing set to make sure that the model could generalise. The first 1,000 points of data from Fig. 7 were used for training (only temperature variation) and the remainder for testing (temperature variation *and* damage).

Figure 9 shows the model fit to the training data; the linear model is clearly incapable of capturing the bilinear dependence on temperature. However, an interesting

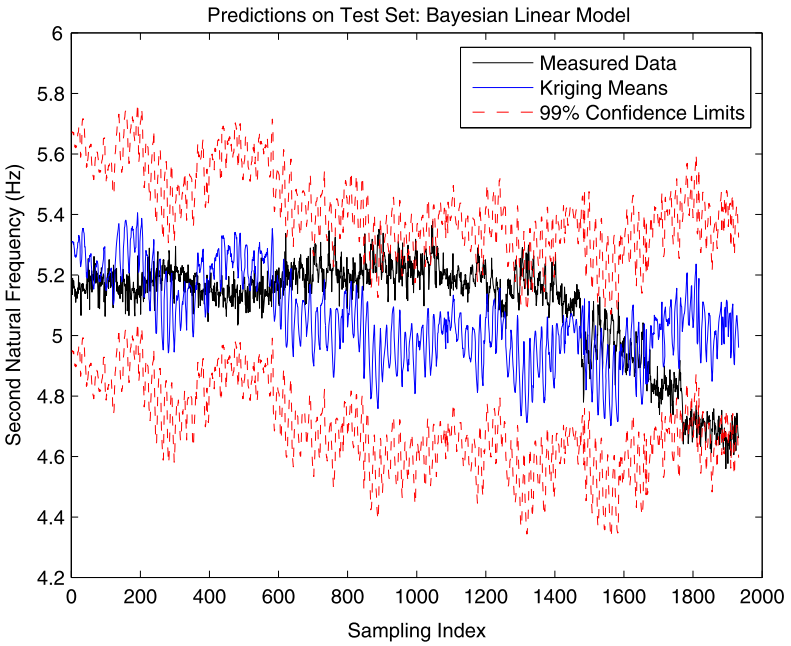


Fig. 10 Bayesian linear model predictions compared to true measurements on testing data set

feature is that the 99 % confidence intervals on the predictions computed by the software do actually encompass the data. Now, recall that the object of the exercise is to *remove* the temperature dependence in order to create a feature for damage detection. The *residual* data for the testing set could be computed by subtracting the model predictions from the data. If the structure were to remain in normal condition throughout the testing period so that temperature changes were the only source of variation, the residual should resemble a stationary noise process with a mean of zero. Confidence intervals could be constructed from the residual values on the training data or could be adopted from the model fit, and then any excursions outside the confidence intervals would potentially indicate damage. As a slightly more informative alternative to plotting residuals, the model predictions with their 99 % confidence limits as compared to the measured data will be shown in this chapter. Damage is indicated when the measured data deviates significantly from the predictions, i.e. goes outside the model confidence intervals. For the Bayesian linear model fitted to the training data, Fig. 10 compares the measured data on the testing set with the model predictions. Although the damage begins to show itself by a monotonic decline in the measured data, the model predictions have so little confidence that there are no excursions outside the 99 % limits. The poor fidelity of the model in capturing the temperature variations has resulted in a residual insensitive to damage. (Within the TGP software, there are a number of methods of computing model predictions, including sampling from the posterior distribution of the model;

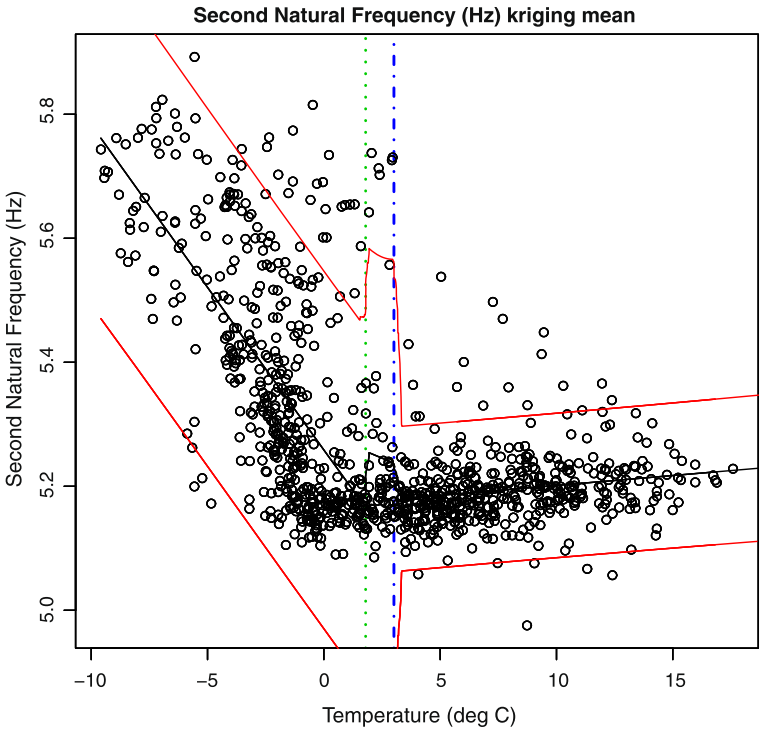


Fig. 11 Bayesian treed linear model of Z24 second natural frequency as a function of temperature: training data. *Vertical dotted* and *dash-dotted lines* indicate the position of switch points

however, one of the simplest methods is to use the kriging means and variances as specified in Eqs. (6) and (7), and this is the approach adopted here.)

The next model fitted was a treed linear model, e.g. a Bayesian CART model. While Fig. 9 shows very clear evidence of the bilinear switching behaviour in the measured data at close to zero Celsius, the variations in the training data between zero and three degrees convince the algorithm that two switching points are needed. This is not an issue in terms of fitting a good predictive model, but it does mean that the model has perhaps not captured the true physics. As one might expect, the treed linear model fits the data much better than the simple linear one, and this is reflected in the much tighter confidence bounds on the training data (Fig. 11). When the model predictions are compared to the measured data on the testing set, the results are far better than those for the linear model; the much higher prediction confidence results in a very clear detection of the damage when the measured data moves outside the confidence bounds not long after initiation of damage (Fig. 12). The important point here is that the complexity of the models in the various regimes is not the issue; more important is the recognition that there *are* different regimes. Another interesting feature can be seen in Fig. 12: the presence of bursts of low prediction confidence at points in the early part of the testing data. The reason for

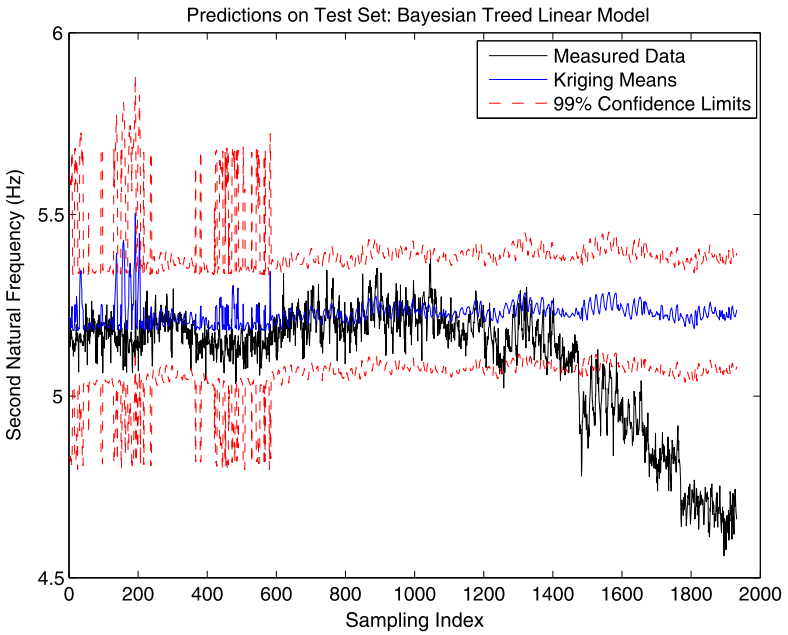


Fig. 12 Bayesian treed linear model predictions compared to true measurements on testing data set

this is that the points correspond to slightly higher temperatures than were present in the training data; the model recognises that it is moving from interpolation to extrapolation and adjusts its confidence accordingly.

For the next exercise, a Bayesian GP model was fitted. This model structure allows the nonlinear dependence on temperature to be modelled, but assumes that it can be captured by a single GP over the entire feature space. The results on the training set are shown in Fig. 13. While the results are better than those of the linear model, they are inferior to those from the treed linear model. The explanation for this lies with the covariance function which determines the smoothness of the predictions for the GP model. At the point of discontinuity in the data, the GP requires a covariance function changing rapidly over short time scales; away from the discontinuity the GP would like a very slowly changing covariance function to reflect the smoothness of the model. The two objectives cannot be met with a single B hyperparameter and so the GP adopts a compromise value; this worsens the prediction confidence and generally expands the confidence interval. The effect of this on damage sensitivity can be seen in the comparison plot of Fig. 14. Although the damage is detected when the test data leaves the prediction confidence intervals, it is detected at a later time than for the treed linear model.

The final model fitted to the data was the treed GP. As there is no real advantage of doing otherwise, the variant of the model allowing switching to a limiting linear model was used. The results on the training data can be seen in Fig. 15. Once again,

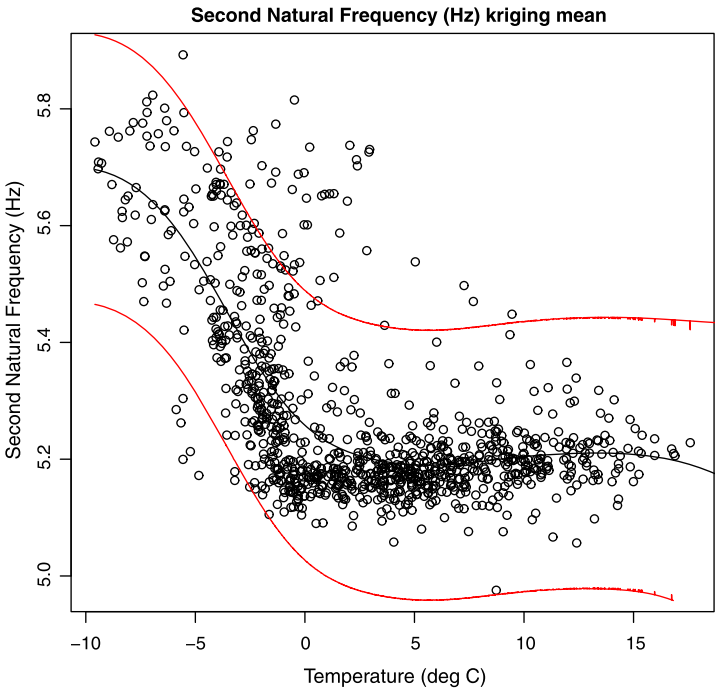


Fig. 13 Bayesian GP model of Z24 second natural frequency as a function of temperature: training data

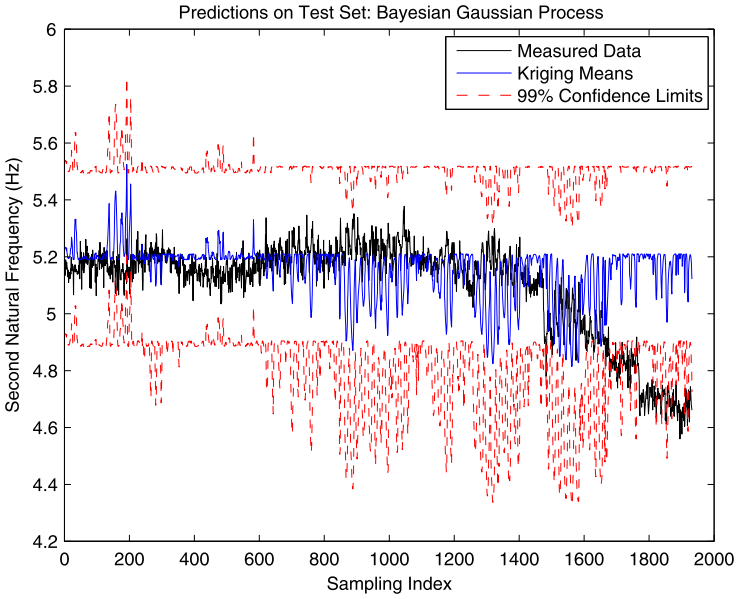


Fig. 14 Bayesian GP model predictions compared to true measurements on testing data set

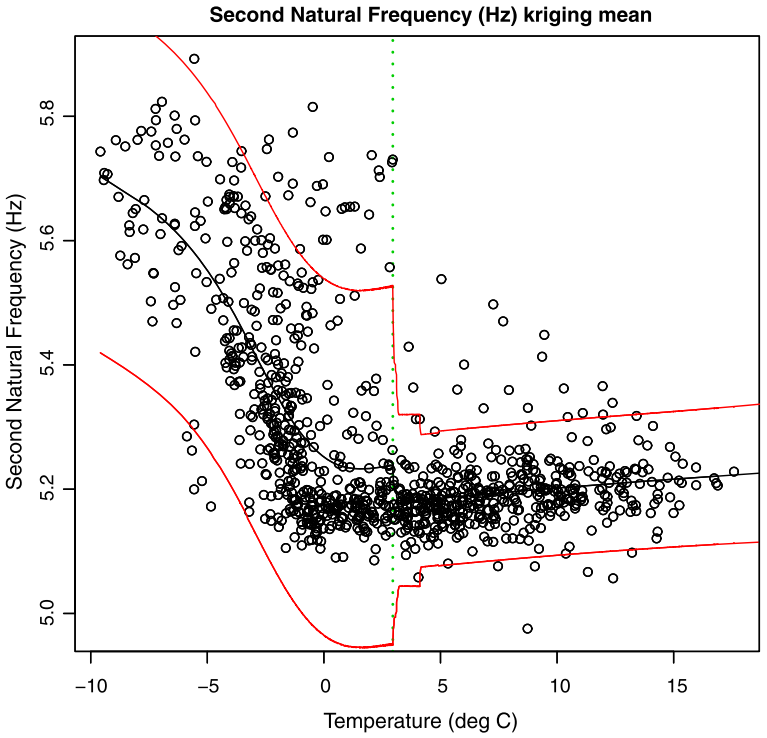


Fig. 15 Bayesian treed GP model of Z24 second natural frequency as a function of temperature: training data. The *vertical dotted line* indicates the position of the switch point

the switching model proves superior; however, in this case there are two additional features of interest for the treed GP. In the first case, the model has recognised that only one switching point is needed. As in the treed linear case, the model switches at a higher point than the freezing point; however, the nonlinear nature of the GP means that the behaviour of the data below that is captured by a single GP. Another interesting aspect of the model is that it switches to a linear model above the switch point; there is nothing to be gained in terms of model evidence from adopting the more complex model. In terms of the damage detection issue, the comparison plot in Fig. 16 shows that the damage is detected as promptly as in the case of the treed linear model but is not significantly different. The treed GP essentially achieves the same result as the treed linear model, but with a simpler partition on the feature space.

5 Conclusions

There is no cause here for lengthy conclusions; the objectives of this chapter have simply been to present a powerful and versatile class of response surface models

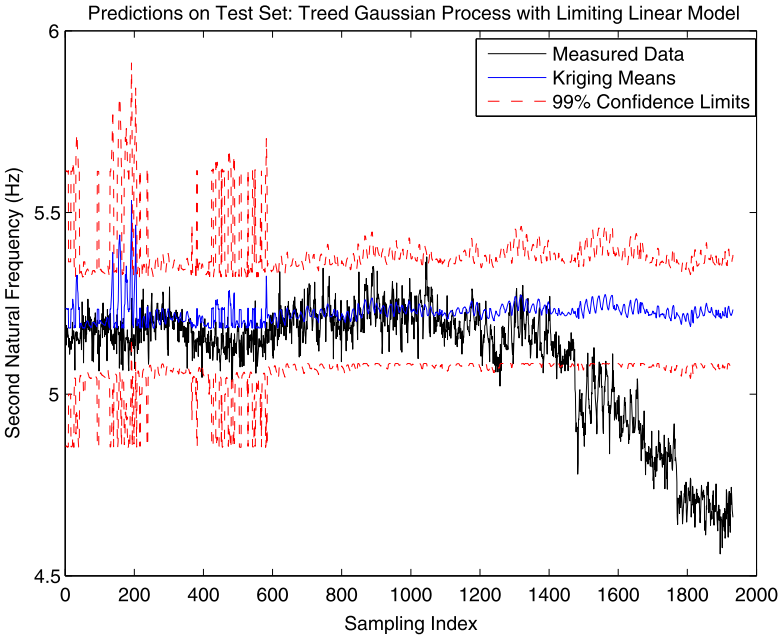


Fig. 16 Bayesian treed GP model predictions compared to true measurements on testing data set

with the capability to switch discontinuously between different regimes and to show the value of the models in the context of vibration data-based SHM. However, there has been an opportunity to touch on some interesting points along the way. The models presented were initially applied here as black-box models—the intention was simply to learn a predictive model from the data. However, the fact that the models can automatically learn the switching points between regimes arguably lifts them into the domain of grey-box models when a physical interpretation can be assigned to the regime boundaries. In the case of the Z24 Bridge, the physics is clear: the asphalt in the deck freezes below zero degrees Celsius and the bridge stiffens. In the case of the Tamar Bridge, the cause of the switching is arguably more obscure; however, it is possible that it is the result of unsteady aerodynamics and as this is a subject of some interest to bridge researchers, identifying and modelling the switching behaviour may well be of value. The models are also arguably superior to other schemes in terms of parsimony. If simple models are allowed over the identified regimes, this means fewer 'coefficients' need to be identified; this in turn reduces the demands on acquiring data and allows models to better generalise away from their training data. Finally, the authors would argue that, if responses are switching discontinuously between regimes, it may be harmful to smooth out the effects by, e.g. fitting overall polynomial models; at the very least, one is not giving proper respect to the physics shown.

Acknowledgements The authors would like to thank their colleague Dr Will Becker of the Joint Research Centre, Ispra for drawing their attention to the TGP software package and for a number of

valuable discussions on the nature of treed and non-treed Gaussian processes. Thanks are also due to Robert Gramacy for providing the community with the TGP package in the first place. In terms of the bridge data discussed here, thanks are due to Dr Ki-Young Koo, formerly of the Department of Civil and Structural Engineering at the University of Sheffield, for helping to provide access to the Tamar Bridge data. The authors would also like to express their gratitude to the researchers on the project SIMCES who conducted the benchmark study on the Z24 Bridge and also made their data freely available. Finally, thanks should also go to Ville Lämsä of Aalto University, Finland for providing the authors with the Z24 data in the form used here.

References

1. Farrar, C.R., Worden, K.: Structural Health Monitoring: A Machine Learning Perspective. Wiley, New York (2013)
2. Farrar, C.R., Baker, W.E., Bell, T.M., Cone, K., Darling, T.W., Duffey, T.A.: Damage characterization and damage detection in the I-40 bridge over the Rio Grande. Los Alamos National Laboratories Report, LA-12767-MS (1994)
3. Sohn, H.: Effects of environmental and operational variability on structural health monitoring. Philos. Trans. R. Soc., Math. Phys. Eng. Sci. **365**, 539–561 (2007)
4. Cross, E.J., Koo, K.-Y., Brownjohn, J.M.W., Worden, K.: Long-term monitoring and data analysis of the Tamar Bridge. Mech. Syst. Signal Process. **35**, 16–34 (2013)
5. Ni, Y.Q., Hua, X.G., Fan, K.Q., Ko, J.M.: Correlating modal properties with temperature using long-term monitoring data and support vector machine technique. Eng. Struct. **27**, 1762–1773 (2005)
6. Ni, Y.Q., Zhou, H.F., Ko, J.M.: Generalization capability of neural network models for temperature-frequency correlation using monitoring data. J. Struct. Eng. **135**, 1290–1300 (2009)
7. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. Chapman & Hall/CRC, London (1984)
8. Chipman, H.A., George, E.I., McCulloch, R.E.: Bayesian CART model search. J. Am. Stat. Assoc. **93**, 935–948 (1998)
9. Chipman, H.A., George, E.I., McCulloch, R.E.: Bayesian treed models. Mach. Learn. **48**, 299–320 (2002)
10. Gramacy, R.B.: Bayesian treed Gaussian process models. Ph.D. Thesis, University of California, Santa Cruz (2005)
11. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2006)
12. Matheron, G.: The intrinsic random functions and their applications. Adv. Appl. Probab. **5**, 439–468 (1973)
13. Kennedy, M.C., Anderson, C.W., Conti, S., O'Hagan, A.: Case studies in Gaussian process modelling of computer codes. Reliab. Eng. Syst. Saf. **91**, 1301–1309 (2006)
14. Becker, W.E.: Uncertainty propagation through large nonlinear models. Ph.D. Thesis, University of Sheffield, Sheffield, UK (2011)
15. Gramacy, R.B.: tgp: an R package for Bayesian nonstationary, semiparametric nonlinear regression and design by treed Gaussian process models. J. Stat. Softw. **19** (2007)
16. Gramacy, R.B., Lee, H.K.H.: Gaussian processes and limiting linear models. Comput. Stat. Data Anal. **53**, 123–136 (2008)
17. Kramer, C., De Smet, C.A.M., De Roeck, G.: Z24 bridge damage tests. In: Proc. 17th Intl. Modal Analysis Conference (IMAC), Kissimmee, Florida (1999)
18. De Roeck, G.: The state-of-the-art of damage detection by vibration monitoring: the SIMCES experience. J. Struct. Control **10**, 127–134 (2003)

19. Yan, A.-M., Kercshen, G., De Boe, P., Golinval, J.-C.: Structural damage diagnosis under varying environmental conditions—part II: local PCA for non-linear cases. Mech. Syst. Signal Process. **19**, 865–880 (2005)
20. Maeck, J., De Roeck, G.: damage assessment using vibration analysis on the Z24 bridge. Mech. Syst. Signal Process. **17**, 133–142 (2003)
21. Mevel, L., Gourset, M., Basseville, M.: Stochastic subspace-based structural identification and damage detection and localisation—application to the Z24 bridge benchmark. Mech. Syst. Signal Process. **17**, 143–151 (2003)
22. Garibaldi, G., Marchesiello, S., Bonisoli, E.: Identification and up-dating over the Z24 benchmark. Mech. Syst. Signal Process. **17**, 153–161 (2003)
23. Kullaa, J.: Damage detection of the Z24 bridge using control charts. Mech. Syst. Signal Process. **17**, 163–170 (2003)
24. Peeters, B., De Roeck, G.: One-year monitoring of the Z24-bridge: environmental effects versus damage events. Earthq. Eng. Struct. Dyn. **30**, 149–171 (2001)

Surrogate Modeling of Stability Constraints for Optimization of Composite Structures

S. Grihon, E. Burnaev, M. Belyaev, and P. Prikhodko

Abstract Problem of aircraft structural components (wing, fuselage, tail) optimization is considered. Solution of this problem is very computationally intensive, since it requires at each iteration a two-level process. First from previous iteration an update step at full component level must be performed in order to take into account internal loads and their sensitivities in the whole structure involved by changes in local geometry. Second numerous local analyzes are run on isolated elements (for example, super stiffeners) of structural components in order to calculate mechanical strength criteria and their sensitivities depending on current internal loads. An optimization step is then performed from combined global-local sensitivities. This bi-level global-local optimization process is then repeated until convergence of load distribution in the whole structure. Numerous calculations of mechanical strength criteria are necessary for local analyzes and results in great increase of the time between two iterations. In this work an effective method for speeding up the opti-

S. Grihon
ESAZO—Optimisation Centre, Rapid Sizing—Optimisation M&T, Airbus Operations SAS, 316, Route de Bayonne, 31060 Toulouse Cedex, France
e-mail: stephane.grihon@airbus.com

E. Burnaev · M. Belyaev · P. Prikhodko
Intelligent Data Analysis Group, DATADVANCE, Pokrovsky blvd. 3 building 1B, 109028, Moscow, Russia

M. Belyaev
e-mail: mikhail.belyaev@datadvance.net

P. Prikhodko
e-mail: pavel.prikhodko@datadvance.net

E. Burnaev (✉) · M. Belyaev · P. Prikhodko
Data Analysis and Modeling Lab, Institute for Information Transmission Problems, Bolshoy Karetny per. 19, Moscow, 127994, Russia
e-mail: burnaev@iitp.ru

E. Burnaev · M. Belyaev · P. Prikhodko
PreMoLab, Moscow Institute of Physics and Technlogy, 141700, 9, Institutskii per., Dolgoprudny, Moscow Region, Russia

S. Koziel, L. Leifsson (eds.), *Surrogate-Based Modeling and Optimization*,
DOI 10.1007/978-1-4614-7551-4_15,

mization process was elaborated. The method uses surrogate models of optimization constraints (mechanical strength criteria) and provides reduction of the structure optimization computational time from several days to a few hours.

Keywords Buckling analysis · Approximation · Mixture of experts · HDA · Composite structure · Surrogate modeling · Optimization

1 Introduction

Aeronautical structures are mainly made of stiffened panels, i.e., thin shells (also called skin) enforced with stiffeners (called frames and stringers) in both the orbital and longitudinal directions. The whole structure is studied by dividing it into elementary parts called super stiffeners, consisting of the theoretical union of a stringer and two half-panels. These basic structures are subject to highly nonlinear phenomena such as buckling, collapse, and damage tolerance.

In order to determine the optimal size of these super stiffeners, static mechanical criteria must be computed using dedicated software based on nonlinear calculations. Thus, the analysis and the dimension estimation of the whole structure is currently computed by running a two-level study: at a global level a finite element (FE) analysis run on the whole FE model provides internal loads applied to each super stiffener; at a local level these loads are used to compute static mechanical criteria. Most of these criteria are formulated using reserve factors (RF): a structure is validated provided all its RFs are greater than one.

Therefore, a detailed design of an aircraft fuselage requires a two-level loop. First, changes from the local geometry, defined at the previous iteration, involve a new internal load distribution in the whole structure; an update step must then be performed to take these changes into account and to compute sensitivities. Second, numerous local analyses are run on isolated super stiffeners to compute mechanical criteria and their sensitivities depending on current internal loads. This bi-level global-local optimization process is then repeated until convergence of the load distribution in the whole structure is achieved.

Local mechanical criteria are computed by local methods, which are used because of the huge dimensionality of the problem ($O(10^4)$ variables and $O(10^5)$ constraints). Local methods require gradients of the constraint functions, defined by static mechanical criteria. These gradients can only be obtained by finite differences. Values of the mechanical strength constraints are computed using dedicated software. A call to this software takes up to a second; as a consequence, the need for finite difference calculations in each of numerous local optimizations greatly increases the time between two update steps.

Therefore, the dimension estimation step in an aircraft development program is a repetitive and time-consuming process. Much time could be saved by using surrogate modeling instead of performing straightforward computing [14, 23]. Thus, the main motivation of this work is a surrogate modeling of buckling analysis in support of composite structure optimization. We want to achieve two goals of great importance for engineers working in the Airbus structural analysis framework: saving

time in the pre-sizing processes and having the advantage of response smoothing, since surrogate models (SMs) provide a continuous and differentiable approximation of RFs that sometimes are not themselves continuous (as is often the case for semi-empirical approaches).

For surrogate modeling of static instability phenomena (the buckling and the collapse of a super stiffener) we used the MACROS software toolkit for surrogate modeling and optimization, developed by DATADVANCE [12].

Finally, the constructed MACROS Surrogate Model (MSM) was embedded into the pre-sizing optimization process of A350XWB composite boxes, realized in a pre-sizing tool COMBOX, for checking the validity of the approximation and its use instead of the corresponding constraint functions in the optimization process. It turned out that MSM allows one to obtain smoother convergence to a reasonable solution in fewer iterations with a smoother distribution of thickness/stringer dimensions and reduces the structure optimization computational time from several days to a few hours.

In the following sections we describe the pre-sizing tool COMBOX (Sect. 2), the surrogate modeling and optimization software toolkit MACROS (Sect. 3), the construction of the MSM for Airbus skill tool (Sects. 3 and 4), and the analysis of the optimization results based on the skill tool and constructed SM (Sect. 5). We end this article with some concluding remarks (Sect. 6).

2 COMBOX: A Pre-Sizing Tool Developed for A350XWB

The COMBOX tool (COMposite BOX pre-sizing) was developed in 2005 to support the pre-sizing of the A350XWB composite wing box (see Fig. 1). It has since been continuously improved and is being applied to all A350XWB boxes: wing, horizontal tail plane, and vertical tail plane.

2.1 COMBOX Sizing Process

The COMBOX sizing process encapsulates the full stress process for a wing box (see Fig. 2):

- Mapping of sizing properties,
- Update of a global finite element model (FEM),
- Calculation of internal loads through a static linear analysis based on the global FEM,
- Calculation of strength responses as reserve factors (RFs) through Airbus skill tools.

These are the usual steps of an airframe structural analysis for pre-sizing.

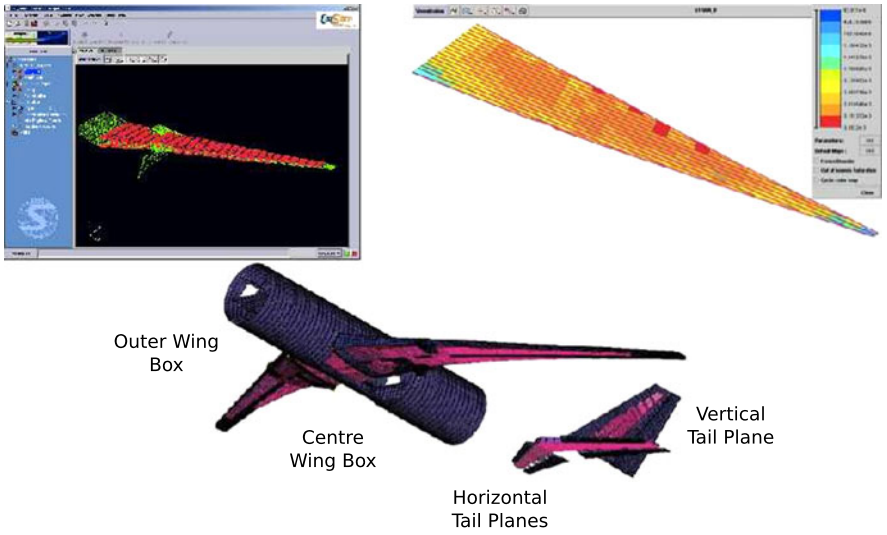


Fig. 1 COMBOX pre-sizing optimization tool is now applied to all A350XWB boxes

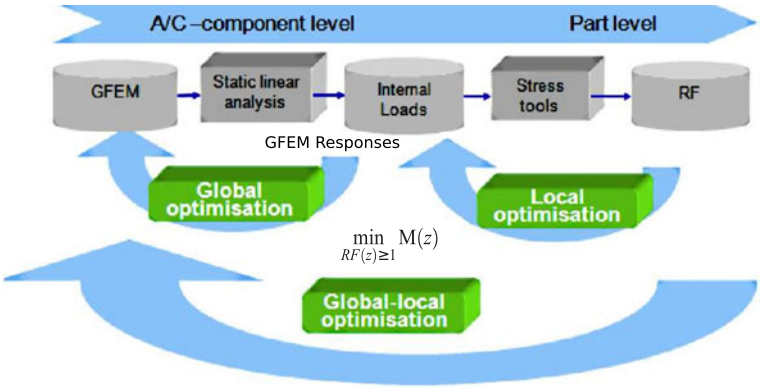


Fig. 2 COMBOX pre-sizing optimization tool is a global-local optimization capability encapsulating the overall stress analysis process

Remarks

- An RF indicates whether the structure is feasible (i.e., has enough strength) with respect to a given mechanical criterion or failure mode. If the RF is greater than one, the structure is feasible. If the RF is less than one, it is not feasible. Therefore, when modeling the dependency of some RF on a vector of design variables $\mathbf{x}$, the highest possible accuracy should be provided for what is called the accuracy domain $\tilde{\mathbf{X}} = \{\mathbf{x} : RF(\mathbf{x}) \in (1-\varepsilon, 1+\varepsilon)\}$, $\varepsilon = 0.2$.
- The simplest example of an RF is a ratio between an allowable stress (for example, material strength) and the applied stress.

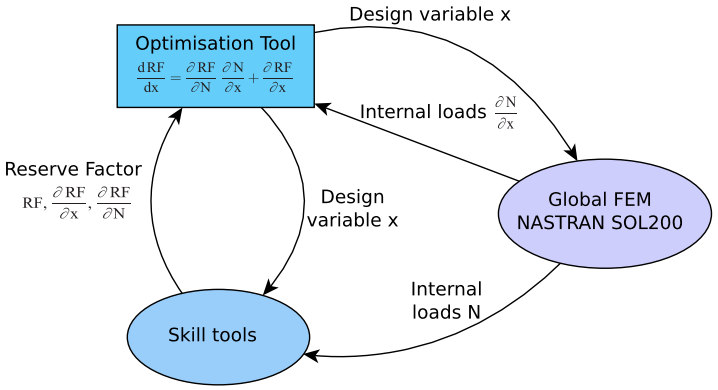


Fig. 3 COMBOX optimization process

- Skill tools are usually analytical semi-empirical tools, which are rather quick and are used for pre-sizing.

2.2 COMBOX Components

COMBOX is based on commercial off-the-shelf software and incorporates four components:

- CAESAM: Software framework from SAMTECH [20] (provides GUI and stress model),
- NASTRAN: Finite element software from MSC [19],
- Skill tools developed by Airbus,
- BOSS Quattro: Optimization software from SAMTECH [21] (provides process manager and optimiser).

2.3 COMBOX Optimization Process

COMBOX is a pre-sizing tool based on numerical optimization (mathematical programming). Therefore, besides sizing calculations (see Sect. 2.1) during optimization process it is necessary to compute the sensitivities of internal loads and RFs and combine them by chain ruling (see Fig. 3). Internal load sensitivities are semi-analytically calculated via the NASTRAN SOL200 module (NASTRAN optimization and sensitivity analysis module). The responses and sensitivities are then sent to the optimization algorithm in BOSS Quattro. CAESAM is mainly used to manage all data, and BOSS Quattro manages the work flow and the optimization process including the sensitivity chain ruling.

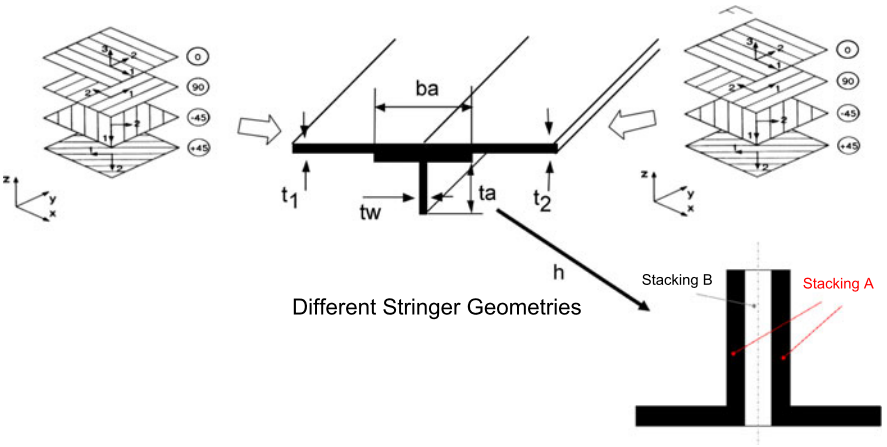


Fig. 4 Illustration of COMBOX design variables

2.4 COMBOX Optimization Problem Formulation

COMBOX is able to address all sizing variables of a composite cover with T-stringers (other stringer sections are possible but not presented here; see also Fig. 4):

- Skin thickness,
- Percentages of standard draping angles: 0 %, 45 %, 90 %,
- T-stringer core and web percentages: 0 %, 45 %, 90 %,
- T-stringer web thickness, core thickness, height, and width.

Bounds are given to these variables to satisfy design rules. Some additional design rules are included like bounds on the As/bt ratio, which represents the ratio of the stringer area to the skin area.

All usual criteria for a composite wing cover sizing are considered (see Fig. 5):

- Local skin buckling and general skin buckling,
- Post-buckling and post-buckling cut-off,
- Skin damage tolerance and stringer damage tolerance,
- Skin reparability and stringer reparability.

RFs are associated to each of these failure modes.

Damage tolerance criteria are there to ensure that the structure can resist small damages. Reparability criteria anticipate some future repairs in the skin (filled hole criteria).

Therefore, the optimization problem can be formulated as

$$M(z) \rightarrow \min_{z \in \mathbf{R}^n} \text{ s.t. } \begin{cases} z_{\text{low}} \leq z \leq z_{\text{up}}, \\ RF_{i,j,k}(N(z), z) \geq 1, \\ \quad i = 1, \ldots, N_e, \, j = 1, \ldots, N_l, \, k = 1, \ldots, N_{\text{fm}}, \\ d_l(z) \geq 1, \end{cases}$$

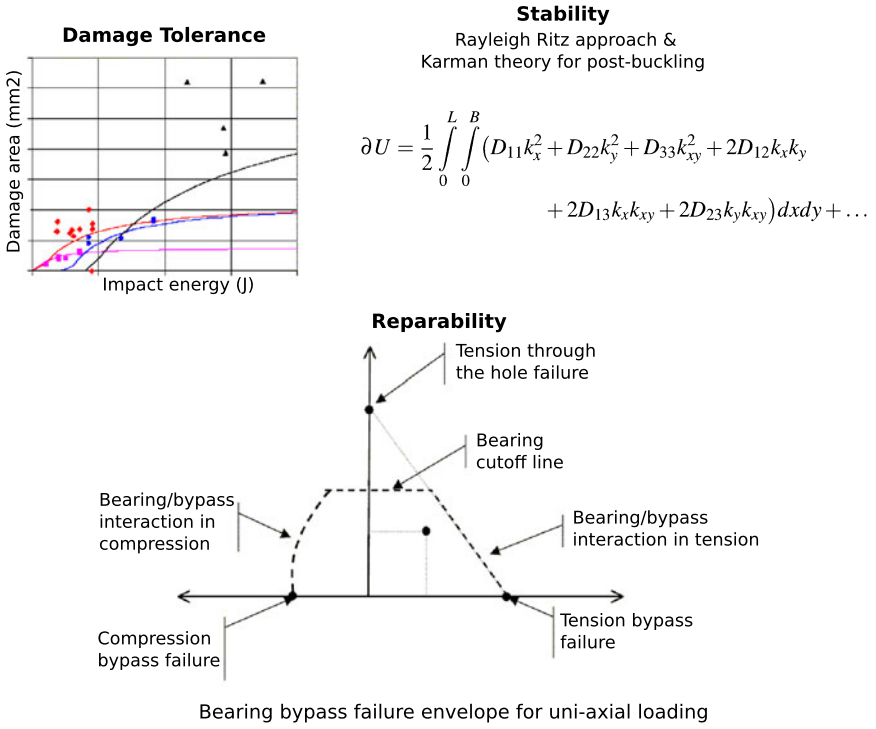


Fig. 5 Illustration of COMBOX strength criteria

where

- The objective function is the mass $M(z)$ of the FEM, independent of percentages, that is to be minimized,
- z is the vector of n optimization variables (skin, stringer thicknesses, dimensions, and percentages),
- $N(z)$ is the vector of internal loads.

The constraints are

- Variable bounds: $z_{\text{low}} \leq z \leq z_{\text{up}}$,
- Strength constraints: $RF_{i,j,k}(N(z), z) \geq 1$, $i = 1, \ldots, N_e$, $j = 1, \ldots, N_l$, $k = 1, \ldots, N_{\text{fm}}$,
- Design constraints: $d_l(z) \geq 1$.

The indexes i, j, k for the strength constraints remind us that there are as many strength constraints as structural elements N_e, external loads N_l, and failure modes N_{fm}. The computational time of the process is mainly contained in the strength analysis due to the high value of $N_e \cdot N_l \cdot N_{\text{fm}}$. On top of that, RF sensitivities are obtained via finite differences; so the number of strength analyses is multiplied by the number of local variables and internal load components (approximately a factor 10).

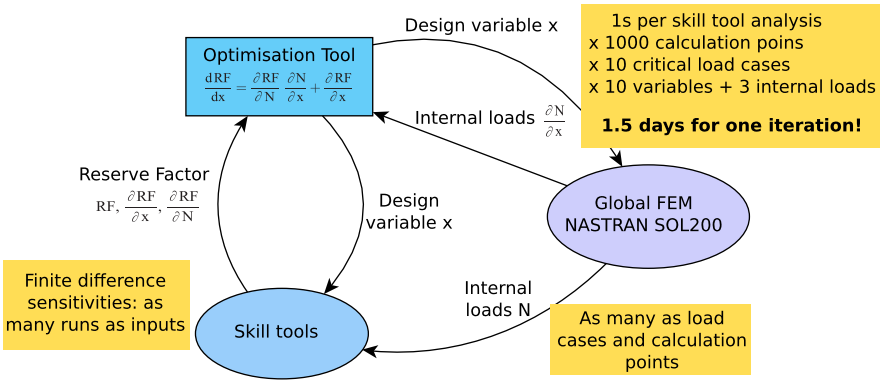


Fig. 6 Computational times in COMBOX

Therefore, even if strength analysis tools for pre-sizing are rather quick (1*s* per element), the total number of calculations is huge and leads from one to five days per iteration with an optimization process usually converging in 20 iterations (see Fig. 6 for details). To save time in the pre-sizing processes and particularly in the COMBOX tool, it is thus necessary to build numerical approximations of the strength tools using surrogate modeling, which is the main goal of this paper. Besides the time reduction, there is also the advantage of response smoothing. Indeed, SMs give a continuous and differentiable approximation of RFs that sometimes are not themselves continuous (as is often the case for semi-empirical approaches). This is also demonstrated in the current study.

3 MACROS: A Surrogate Modeling and Optimization Software Toolkit

MACROS is a software toolkit for

- Intellectual data analysis, and
- Multidisciplinary optimization,

developed by DATADVANCE [12]. It provides proprietary and state-of-the-art data analysis and optimization techniques.

The MACROS toolkit consists of Generic Tools (GTs) for Dimension Reduction, Important Variable Extraction, Design of Experiments, Approximation, Data Fusion, and Optimization.

GT for Dimension Reduction includes unsupervised and supervised (feature extraction) techniques for automatic reparameterization of an object's description with a smaller number of parameters.

GT for Important Variable Extraction includes techniques for sensitivity analysis necessary for ranking the available parameters with respect to their influence on the given response function and selecting the most important ones.

GT for Design of Experiments enables systematic and efficient analysis of the design space by using classical and advanced methods (full factorial, optimal Latin hypercube, Halton and Sobol sequences, etc.) as well as specially designed adaptive techniques.

GT for Approximation allows automatic construction of fast-running data-based SMs using best-in-class predictive modeling techniques. The tool includes built-in robustness and accuracy assessment, control of SM smoothness, etc., and is efficient for small and huge data samples in low and high dimensions.

GT for Data Fusion allows approximating data of variable fidelities. The tool operates like GT for Approximation, but assumes that the response function is represented by two types of data: scarce high-fidelity data and abundant low-fidelity data. The tool then constructs an enhanced approximation of the high-fidelity model taking into account the abundant low-fidelity data.

GT for Optimization includes efficient state-of-the-art optimization methods to solve various problems (large-scale, linear/nonlinear, unconstrained/constrained, single/multi-objective, and stochastic).

Adaptive and automatic selection of the best method for a given problem on the basis of specially designed decision trees opens up elaborated methods for use by people who interact with problems on the engineering rather than the mathematical level.

4 Construction of MACROS Surrogate Model

Let us describe the following in this section:

- Proposed methodology for surrogate model (SM) construction, based on mixture of experts framework and used for construction of MACROS Surrogate Model (MSM),
- New High Dimensional Approximation (HDA) algorithm, implemented in GT for Approximation (GT Approx) and used for construction of experts (approximations in local regions),
- Differences and similarities between the proposed and already existing approaches for SM construction including results of computational experiments.

Construction of the SM is necessary for obtaining a more computationally efficient approximation of the original dependency. Therefore, let us formulate engineering statement of the approximation problem and then formulate requirements, which we impose on the SM.

4.1 Approximation Problem Statement

Let us denote by

$$S_{\text{learn}} = \left\{ (\mathbf{x}_i, y_i), i = 1, \ldots, N_{\text{learn}} \right\} \tag{1}$$

points generated independently randomly such that there is some unknown functional dependency $y_i = f(\mathbf{x}_i)$ between the output value (output) $y_i \in \mathbf{Y} \subset \mathbf{R}^1$ and the input vector (input) $\mathbf{x}_i \in \mathbf{X} \subset \mathbf{R}^p$.

The SM construction problem statement is to construct an approximation (approximator, approximating model) $\hat{f}(\mathbf{x}) = \hat{f}(\mathbf{x}|S_{\text{learn}})$ for the given dependency $f(\mathbf{x})$ using learning sample S_{learn} such that for all $\mathbf{x} \in \mathbf{X}$ (not only for $\mathbf{x} \in S_{\text{learn}}$) the following approximate equality holds:

$$\hat{f}(\mathbf{x}) \approx f(\mathbf{x}), \tag{2}$$

i.e., the approximator $\hat{f}(\mathbf{x})$ has good generalization ability and recovers the given dependency with good accuracy.

Equation (2) is considered to be fulfilled if on independent test set $S_{\text{test}} = \{(\mathbf{x}_j, y_j), j = 1, \ldots, N_{\text{test}}\}$ the value of the error

$$\hat{er}_{S_{\text{test}}}(f, \hat{f}) = \frac{1}{N_{\text{test}}} \sum_{j=1}^{N_{\text{test}}} \left(y_j - \hat{f}(\mathbf{x}_j)\right)^2 \tag{3}$$

is small (accuracy is high).

In order for the criterion (3) of approximation quality to make sense, the input vectors from the samples S_{learn} and S_{test} should be generated by the same distribution and distributed in $\mathbf{X}$ sufficiently densely.

In practice, when constructing an SM $\hat{f}(\mathbf{x})$, additional requirements and data generation source properties often should be taken into account.

4.1.1 Specific Requirements on Accuracy

There can exist different requirements on the accuracy of the SM in different domains of the design space $\mathbf{X}$. For example, when constructing an MSM for the considered stability constraints approximation problem, high accuracy of prediction should be provided in the domain $\tilde{\mathbf{X}} \subset \mathbf{X}$, where $\tilde{\mathbf{X}} = \{\mathbf{x} : f(\mathbf{x}) \in (1 - \varepsilon, 1 + \varepsilon)\}$, $\varepsilon = 0.2$. Due to this requirement, it is necessary to construct approximation only in the domain $\tilde{\mathbf{X}}$ using the subsample $\tilde{S} = \{(\mathbf{x}, y) \in S_{\text{learn}} : \mathbf{x} \in \tilde{\mathbf{X}}\}$ and then *glue* it with approximation, constructed in the domain $\mathbf{X} \setminus \tilde{\mathbf{X}}$ using the subsample $S_{\text{learn}} \setminus \tilde{S}$. Since the variation of the approximable function $f(\mathbf{x})$ is smaller in the domain $\tilde{\mathbf{X}}$, than in the whole design space $\mathbf{X}$, then this approach will allow to construct a more accurate approximation for $\mathbf{x} \in \tilde{\mathbf{X}}$.

4.1.2 Spatial Inhomogeneity of the Sample

When decomposing the design space and selecting domains (see Sect. 4.1.1) corresponding to different requirements on the accuracy of the SM, it can happen that the majority of these domains can be represented as the unions of some disconnected

sets. Subsamples, corresponding to the selected domains, will also be some unions of clusters of points.

It is obvious that the global SM will have poor accuracy if it is to be constructed using a sample that is some union of several clusters of points. Thus it is reasonable to perform a preliminary decomposition of such a sample into several homogeneous subsamples, each of which is located in the connected subdomain of the design space. Then, using each subsample, a local approximation (expert) is constructed in the corresponding subdomain.

4.1.3 Redundancy in Data

It can also happen that the set of input parameters is redundant in one of the two (or even in both) senses:

- Input parameters can be dependent. In the simplest case it means that several input parameters are correlated.
- It may be that a function does not depend on all input parameters. In this situation, two main scenarios are worth considering:
  - the function weakly depends on several inputs,
  - the function depends not on initial inputs, but on their projection onto some linear subspace of smaller dimension.

Detection and removal of such redundancies in general allows us to significantly improve the quality of the constructed SM.

4.2 Methodology for Surrogate Model Construction

Let some sample S_{learn} (1) be given. Also, let us use GT Approx for construction of an approximation model $y = g(\mathbf{x}, \theta)$ based on the given sample S_{learn}. By *construction of the approximation model* we mean selecting some element $g(\cdot, \theta)$ from the predefined parametric family G by tuning parameters θ such that approximation is optimal with respect to criterion (3).

An elaborated approach for construction of the SM can be described as follows:

1. Methods to remove redundancy. In order to remove redundancy from the input parameters, methods for important variables extraction, dimension reduction, and feature extraction are used. Application of these methods for preprocessing the data will not be considered here further, since these methods are not used for constructing an SM in the considered applied problem. The problem statement and a detailed description of the method for effective dimension reduction are given in [8].

2. Decomposition of the design space into domains $\mathbf{X} = \bigcup_{j=0}^{N_y-1} \mathbf{X}_j$, corresponding to different ranges of the output. This decomposition is useful, since for different ranges of the output we need to provide different approximation accuracies. A detailed description of the procedure is given in Sect. 4.2.1.
3. Decomposition of the domains $\mathbf{X}_j$ into connected subdomains $\mathbf{X}_{j,k}$, corresponding to more regular behavior of the approximable dependency. A detailed description of the process is given in Sect. 4.2.2.
4. Construction of the approximators using the subsamples $S_{j,k} = \{(\mathbf{x}, y) \in S_{\text{learn}} : \mathbf{x} \in \mathbf{X}_{j,k}\}$.
5. Construction of the classifier that *estimates the proximity* from the given point $\mathbf{x}$ to subdomains $\mathbf{X}_j$. A detailed description of the process is given in Sect. 4.2.3.
6. Construction of the final SM by *gluing* obtained approximators. A detailed description of the process is given in Sect. 4.2.3.

4.2.1 Decomposition of the Design Space into Domains Based on Output Values

Decomposition of the design space into domains based on output values can be described as follows:

1. Let $y_{\min}$ and $y_{\max}$ be an upper and a lower bounds on the output value $y = f(\mathbf{x})$. The interval of output variation $y \in [y_{\min}, y_{\max}]$ is partitioned into N_y subintervals, i.e., $[y_{\min}, y_{\max}] \in \bigcup_{j=0}^{N_y-1} [y^{2j}, y^{2j+1}]$, where $y^0 = y_{\min}$, $y^{2N_y-1} = y_{\max}$, and $y^{2j} < y^{2j+1}$, $y^{2j+2} < y^{2j+1}$, $j = 0, 1, \ldots, N_y - 1$. Conditions on the ends of the subintervals provide nonempty intersections of these subintervals. This allows us to provide smooth *gluing* of the corresponding approximators (see Sect. 4.2.3). Selection of the decomposition is done, for example, according to accuracy requirements on the SM, imposed by the subject domain. The decomposition $\mathbf{X} = \bigcup_{j=0}^{N_y-1} \mathbf{X}_j$ of the design space corresponds to such a partition, where $\mathbf{X}_j = \{\mathbf{x} : f(\mathbf{x}) \in [y^{2j}, y^{2j+1}]\}$.
2. The sample S_{learn} is partitioned into N_y subsamples $S_{\text{learn}} = \bigcup_{j=0}^{N_y-1} S_j$ such that $S_j = \{(\mathbf{x}, y) \in S_{\text{learn}} : y \in [y^{2j}, y^{2j+1}]\}$. An approximator $f^j_{\text{approx}}(\mathbf{x})$ is constructed using each subsample S_j. This approximator can be some model from G (e.g., it can be constructed using GT Approx), or it can have a more complex structure; see Sect. 4.2.2.

We should note that the described decomposition of the design space into domains based on output values is not only useful if there are different requirements on the accuracy of the SM in different regions of the design space $\mathbf{X}$. In fact, if the function $f(\mathbf{x})$ is significantly spatially inhomogeneous, then the variability of the function $f(\mathbf{x})$ in the domain $\mathbf{X}_j \subset \mathbf{X}$ is significantly lower than in all the design space $\mathbf{X}$. Thus, if approximations are constructed for domains $\mathbf{X}_j \subset \mathbf{X}$ and are *glued*, then a more accurate SM can be obtained compared to the global SM, constructed at once for all the design space $\mathbf{X}$.

4.2.2 Decomposition of the Design Space into Domains Based on Input Values

In this subsection it is described how to additionally decompose the input design space into subdomains with more regular behavior of the approximable function.

An approximator $f^j_{\text{approx}}(\mathbf{x})$ for $\mathbf{x} \in \mathbf{X}_j$ (see Sect. 4.2.1) can be constructed using the sample S_j as an approximation model from G (by applying GT Approx to the sample S_j). However, if the sample S_j is significantly spatially inhomogeneous or even is represented by several separated clusters of points, then the approximator, constructed on the basis of this inhomogeneous sample, will in general have lower accuracy compared to an approximator constructed using a more uniform sample. In order to increase the accuracy, it is proposed to

- Additionally decompose the domains $\mathbf{X}_j$ into connected subdomains $\mathbf{X}_j = \bigcup_{k=1}^{N_x} \mathbf{X}_{j,k}$, such that
  - subsamples of the sample S_j, belonging to these connected subdomains, are more homogeneous,
  - within the subdomains $\mathbf{X}_{j,k}$ approximable dependency $f(\mathbf{x})$ has actually more regular behavior.
- Construct a separate local approximator $g_{j,k} \in G$ for each subdomain $\mathbf{X}_{j,k}$; then the final approximating model $f^j_{\text{approx}}(\mathbf{x})$ is represented as a continuous mixture of these local models.

For constructing the decomposition $\mathbf{X}_j = \bigcup_{k=1}^{N_x} \mathbf{X}_{j,k}$ on the basis of the sample S_j, $j = 0, \ldots, N_y - 1$, it is proposed to use a Gaussian mixture model (GMM, see [15]). It is assumed that the points $(\mathbf{x}, y) \in S_j$ are generated according to the model

$$\text{Law}(\mathbf{x}, y) = \sum_{k=1}^{N_x} \alpha^j_k \mathcal{N}\left(\mu^j_k, \Theta^j_k\right), \quad \sum_{k=1}^{N_x} \alpha^j_k = 1, \ \alpha^j_k > 0, \tag{4}$$

where μ^j_k and Θ^j_k are the mean vector and the covariance matrix for the k-th normal distribution of the GMM, generating j-th sample S_j. Also, the unconditional distribution of the input vector $\mathbf{x} \in \mathbf{X}_j$ has the form $\text{Law}(\mathbf{x}) = \sum_{k=1}^{N_x} \alpha^j_k \mathcal{N}(\mu^j_{k,x}, \Theta^j_{k,x})$, where $\mu^j_{k,x}$ and $\Theta^j_{k,x}$ are the subvector of the mean vector μ^j_k and the submatrix of the covariance matrix Θ^j_k, respectively.

Parameters of the GMM are estimated using the sample S_j by the standard EM algorithm [15], and in the framework of the GMM model additional clustering of the sample S_j is done. Note that since for clustering we use not only input values but also output values, then we take into account possible spatial inhomogeneity of the function in the domain $\mathbf{X}_j$. Then estimated parameters of the GMM in fact will provide the decomposition $\mathbf{X}_j = \bigcup_{j=1}^{N_x} \mathbf{X}_{j,k}$. Indeed, let us define the Mahalanobis distance from the center of the k-th cluster to the point $\mathbf{x}$ according to the formula $d_{j,k}(\mathbf{x}) = (\mathbf{x} - \mu^j_{k,x})^T (\Theta^j_{k,x})^{-1} (\mathbf{x} - \mu^j_{k,x})$; then the set $\mathbf{X}_{j,k} = \{\mathbf{x} \in \mathbf{X}_j : d_{j,k}(\mathbf{x}) \le \chi^2_{97\%}(p)\}$, where $\chi^2_{97\,\%}(p)$ is a 97 % quantile of the distribution χ^2 with p degrees

of freedom. Local approximators $g_{j,k} \in G$ for each subdomain $\mathbf{X}_{j,k}$ are constructed using subsamples $S_{j,k} = \{(\mathbf{x}, y) \in S_{\text{learn}} : \mathbf{x} \in \mathbf{X}_{j,k}\}$, $k = 1, \ldots, N_x$ and GT Approx.

Let us estimate the weight characterizing to what extent the point $\mathbf{x}$ *belongs* to the k-th cluster according to the formula

$$w(k|\mathbf{x}, j) = \frac{1}{2}\left(\tanh\left(1 - 2\frac{d_{j,k}(\mathbf{x}) - \chi^2_{97\%}(p)}{\chi^2_{99\%}(p) - \chi^2_{97\%}(p)}\right) + 1\right), \tag{5}$$

where $\chi^2_{99\%}(p)$ and $\chi^2_{97\%}(p)$ are 99 % and 97 % quantiles of the distribution χ^2 with p degrees of freedom, respectively. In the framework of the model GMM (4), the classifier, which estimates the extent to which the point $\mathbf{x}$ belongs to the k-th cluster, can be constructed according to the formula

$$\hat{w}(k|\mathbf{x}, j) = \frac{w(k|\mathbf{x}, j)}{\sum_r w(r|\mathbf{x}, j)}. \tag{6}$$

The final prediction is obtained using smooth *gluing* of the local models $g_{j,k}$ for all clusters to which the point belongs and is calculated according to the formula

$$f^j_{\text{approx}}(\mathbf{x}) = \sum_{k=1}^{N_x} \hat{w}(k|\mathbf{x}, j) g_{j,k}(\mathbf{x}), \tag{7}$$

where $g_{j,k}(\mathbf{x})$ is a local approximator in the k-th cluster, constructed using the subsample $S_{j,k}$.

In fact, when constructing MSM, we initially tried to use weights equal to the posterior probabilities $\hat{w}(k|\mathbf{x}, j) = \mathrm{P}(k|\mathbf{x}, j)$ of the point $\mathbf{x}$ to belong to the corresponding clusters. Of course, for any point $\mathbf{x}$ there exist such clusters that are located far from it, but nonetheless the corresponding posterior probabilities $\hat{w}(k|\mathbf{x}, j) = \mathrm{P}(k|\mathbf{x}, j)$ are not zero. This means that predictions from local approximators, constructed for these clusters, are taken into account in the mixture (7), thus introducing the error into prediction.

Experiments showed that an additional significant increase in accuracy of MSM can be obtained by cutting the weights according to the distance to the cluster; i.e., we define the weight according to the formula

$$\tilde{w}(k|\mathbf{x}, j) = \begin{cases} \hat{w}(k|\mathbf{x}, j) & \text{if } d_{j,k}(\mathbf{x}) \le \chi^2_{99\%}(p), \\ 0 & \text{else.} \end{cases} \tag{8}$$

However, it is obvious that such an approach introduces discontinuities into MSM, due to which usage of MSM in the optimization process is impossible.

Thus, in order to smooth discontinuities and at the same time additionally penalize predictions (by decreasing the corresponding weights) which correspond to clusters lying far away from the considered point $\mathbf{x}$, we have introduced the approach based on the formula (5). Experiments showed that such an approach is more robust/accurate compared to other possible approaches.

4.2.3 Construction of the Classifier and Calculation of the Output Value of the Surrogate Model

In order to select approximating models $f^j_{\text{approx}}(\mathbf{x})$, $j = 0, \ldots, N_y - 1$ for calculation of the prediction $\hat{f}(\mathbf{x})$, it is proposed to use the classifier $f_{\text{class}}(\mathbf{x}) \in G$ that is a usual approximator, constructed on the basis of the whole sample S_{learn} by GT Approx. In this case the prediction $\hat{f}(\mathbf{x})$ for the given input vector $\mathbf{x}$ is calculated as follows:

- Calculate the value $f_{\text{class}}(\mathbf{x})$. Let us denote by $\#(A)$ the capacity of the set A, $J(\mathbf{x}) = \{j \in (0, \ldots, N_y - 1) : f_{\text{class}}(\mathbf{x}) \in [y^{2j}, y^{2j+1}]\}$. According to the conditions imposed on the decomposition $[y_{\min}, y_{\max}] \in \bigcup_{j=0}^{N_y-1} [y^{2j}, y^{2j+1}]$ of the output range (see Sect. 4.2.1), it holds that $\#(J(\mathbf{x})) \in \{1, 2\}$.
- If $\#(J(\mathbf{x})) = 1$, then calculate the prediction according to the formula $\hat{f}(\mathbf{x}) = f^{J(\mathbf{x})}_{\text{approx}}(\mathbf{x})$.
- If $\#(J(\mathbf{x})) = 2$, then, defining by $j \in J(\mathbf{x})$ the smallest of the two index values from the set $J(\mathbf{x})$, calculate the prediction according to the formula

$$\hat{f}(\mathbf{x}) = f^j_{\text{approx}}(\mathbf{x}) \left(1 - v\left(\frac{f^{j+1}_{\text{approx}}(\mathbf{x}) - y^{2j+2}}{y^{2j+1} - y^{2j+2}}\right)\right) + f^{j+1}_{\text{approx}}(\mathbf{x}) v\left(\frac{f^{j+1}_{\text{approx}}(\mathbf{x}) - y^{2j+2}}{y^{2j+1} - y^{2j+2}}\right),$$

  where $v(x) = 3x^2 - 2x^3$. This definition of the weight function $v(x)$ ensures smooth *gluing* of the models $f^j_{\text{approx}}(\mathbf{x})$ and $f^{j+1}_{\text{approx}}(\mathbf{x})$.

4.2.4 Positioning of the Proposed Methodology for Surrogate Model Construction Among Other Similar Methodologies

As can be seen from reviews [14, 26], the construction of an SM is usually considered to be just a solution of an approximation problem using conventional approximation methods such as (see [5, 11, 15]):

- Kriging (Gaussian process regression),
- Artificial neural networks (ANNs),
- Radial basis functions (RBFs),
- Support vector regression (SVR),
- Multivariate nonparametric regression,
- Polynomial regression, etc.

However these standard methods cannot provide sufficient accuracy, especially when approximating spatially inhomogeneous functions.

Therefore, in [4] a mixture of experts based methodology called IMAGE (Improved Metamodeling Approximation through Gaussian mixture of Experts) for SM

construction was described. However, we have the following significant differences between the approach of [4] and the elaborated methodology.

1. Before decomposing the input design space using the GMM and EM algorithm (as in [4]), it is proposed to:
   - perform decomposition of the design space into domains based on output values. Such an approach is allowed to ensure more accurate approximation in the region $\tilde{\mathbf{X}} = \{\mathbf{x} : f(\mathbf{x}) \in (1-\varepsilon, 1+\varepsilon)\}$, $\varepsilon = 0.2$.
   - perform effective dimension reduction, for which an efficient algorithm based on Gaussian processes was developed; see [8].
2. Instead of using posterior probabilities as weights in mixture (7) (as in [4]), we use weights directly based on the Mahalanobis distance from the considered point $\mathbf{x}$ to clusters combined with a sigmoid activation function. This definition allows the additional penalization of terms in (7) that correspond to clusters located far away from the considered point $\mathbf{x}$.
3. Instead of using standard approximation techniques (RBF, SVR, etc.), as was done in [4], the powerful High Dimensional Approximation (HDA) technique, implemented in GT Approx, is applied. A short description of the HDA algorithm and its comparison with conventional approximation methods are given in Sect. 4.3.

The IMAGE surrogate modeling approach, based on a mixture of experts (standard approximation techniques are used as experts, such as RBF, ANN, SVR, etc.) and described in [4], was extensively compared, using data from the considered problem (see Sects. 1 and 2), with the global approximation algorithm HDA implemented in GT Approx (see Sect. 4.3). It turned out that:

- HDA provides better accuracy of approximation than conventional methods (see Sect. 4.3.4 for details),
- HDA global approximation provides better accuracy of approximation than the IMAGE SM, composed of local experts which were trained using conventional methods (see Sect. 4.3.5 for details).

Therefore it is reasonable to combine the mixture of experts approach for working with spatially inhomogeneous functions with the good approximation properties of HDA. This is done in the proposed framework for SM construction and allows us to obtain an efficient and accurate solution for the considered problem. See Sect. 5 for details.

4.3 High-Dimensional Approximation

As was mentioned in Sect. 4.2 (see also Sect. 4.2.2), MACROS GT Approx is used to construct approximation model $g(\mathbf{x})$ using the given sample S_{learn} and best-in-class predictive modeling techniques. For the considered problem, described in

Sects. 1 and 2, the decision tree built in GT Approx automatically selects the High Dimensional Approximation (HDA) method, which is efficient in the case of huge data samples and high input dimensions. The HDA method is briefly described as follows.

HDA approximator $g(\mathbf{x})$ (see also [9] for details) consists of several basic approximators $g_i(\mathbf{x}), i = 1, 2, \ldots$, which are iteratively constructed and integrated into $g(\mathbf{x})$ using a specially elaborated boosting algorithm, until the accuracy of approximator $g(\mathbf{x})$ stops to increase. In fact,

$$g(\mathbf{x}) = \frac{1}{B} \sum_{k=1}^{B} g_k(\mathbf{x}), \tag{9}$$

where the number B of basic approximators $g_k(\mathbf{x}), k = 1, 2 \ldots, B$ is also estimated by the boosting algorithm. Each $g_k(\mathbf{x}), k = 1, 2, \ldots$ is trained on the modified sample $S_k = \{(\mathbf{x}_i, \widetilde{y}_{i,k}), i = 1, \ldots, N_{\text{learn}}\}$, where $\widetilde{y}_{i,k}$ is some function of $(\mathbf{x}_i, y_i) \in S_{\text{learn}}$ and $\{\hat{y}_{i,j} = g_j(\mathbf{x}_i), j = 1, 2, \ldots, k-1\}$. See [7] for details.

In turn, basic approximators $g_k(\mathbf{x}), k = 1, 2, \ldots, B$ are represented as some averages

$$g_k(\mathbf{x}) = \frac{1}{M_k} \sum_{l=1}^{M_k} h_{k,l}(\mathbf{x}), \quad k = 1, 2, \ldots, B$$

of elementary approximators $h_{k,l}(\mathbf{x}), l = 1, \ldots, M_k, k = 1, 2, \ldots, B$, obtained using multistart on their parameters. The value of M_k is estimated by the HDA training algorithm. An elementary approximator model is described in the next subsection.

4.3.1 Elementary Approximator Model

A linear expansion of parametric functions from the dictionary is used as an elementary approximator model in HDA; i.e., the elementary approximator has the form

$$h(\mathbf{x}) = \sum_{j=1}^{q} \alpha_j \psi_j(\mathbf{x}), \tag{10}$$

where $\psi_j(\mathbf{x}), j = 1, \ldots, q$ are some parametric functions. Three main types of parametric functions are used (the justification for using these basis functions is given in [15]), namely:

1. Sigmoid basis function $\psi_j(\mathbf{x}) = \sigma(\sum_{i=1}^{p} \beta_{j,i} x_i)$, where $\mathbf{x} = (x_1, \ldots, x_p)$, $\sigma(z) = \frac{e^z - 1}{e^z + 1}$. In order to model sharp features a different parameterization can be used, namely,

$$\psi_j(\mathbf{x}) = \sigma\left(\left|\sum_{i=1}^{p} \beta_{j,i} x_i\right|^{\sigma(\alpha_j)+1} \operatorname{sign}\left(\sum_{i=1}^{p} \beta_{j,i} x_i\right)\right),$$

where parameter α_j is adjusted independently of parameters $\beta_j = (\beta_{j,1}, \ldots, \beta_{j,p})$. The essence is that for big negative values of α_j the function $\psi_j(\mathbf{x})$ behaves like a step function.

2. Gaussian functions $\psi_j(\mathbf{x}) = \exp(-\|\mathbf{x} - d_j\|_2^2/\sigma_j^2)$.
3. Linear functions $\psi_j(\mathbf{x}) = x_j$, $j = 1, 2, \ldots, p$, $\mathbf{x} = (x_1, \ldots, x_p)$.

Thus the index set $J = \{1, \ldots, q\}$ can be decomposed into three parts $J = J_{\text{lin}} \cup J_{\text{sigmoid}} \cup J_{\text{GF}}$, where $J_{\text{lin}} \subseteq \{1, \ldots, p\}$ corresponds to the linear part (linear functions), and J_{sigmoid} and J_{GF} correspond to sigmoid and Gaussian functions, respectively. Therefore, in order to fit the model (10) to the data, we should choose the number and type of functions q, and estimate their parameters by minimizing the mean-square error (performance function) on the learning set.

4.3.2 Training of Elementary Approximator Model

Training of the elementary approximator model (10) consists of the following steps:

1. The parameters of the functions from the dictionary are initialized (see description of the algorithm in [3]). An initial number of functions is selected with redundancy.
2. The model selection is made, i.e., values of q, $\#(J_{\text{sigmoid}})$, and $\#(J_{\text{GF}})$ are estimated, and redundant functions are deleted (see the descriptions of algorithms in [2, 7, 9]).
3. The parameters of the approximator are tuned using a hybrid algorithm based on regression analysis and gradient optimization methods. At each step of this algorithm:

   - parameters of linear decomposition (10) are estimated using ridge regression with adaptive selection of regularization parameter, while parameters of the functions from the decomposition are kept fixed,
   - parameters of the functions from the decomposition are tuned using a trust region based gradient method, while parameters of linear decomposition (10) are kept fixed.

   Additional details of the algorithm can be found in [1, 9].

4.3.3 Positioning of HDA Among Other Approximation Methods

The approximation problem appears in many applications, each of which imposes its specific requirements on the accuracy and properties of the approximator $\hat{f}(\mathbf{x})$. Also, depending on the nature of the data source, the corresponding learning sample can have different characteristics (for example, the size of the learning sample N_{learn} and input dimension p). It is natural that the selection of the approximation method should depend on both the requirements of the considered subject domain and the specific properties of the data. For example, a distinctive feature of many problems

in biology is a very high input dimensionality $N_{\text{learn}} \ll p$, so application of nonlinear approximation methods for such problems is senseless. In the current work we consider an approximation problem in the framework of surrogate modeling, which also imposes its own requirements on the approximation method. The main peculiarity of data samples from this applied domain is a large sample size (N_{learn} can be up to several hundred thousand points), with input dimension p usually of size from 3 up to 50. Typically, the unknown function f has a complex nonlinear structure, and the approximation quality should be rather high. Along with that, approximation $\hat{f}$ should be smooth enough and have continuous derivatives, since very often an obtained approximation is then used as a cost function (or as an objective) in some optimization process.

One of the most widespread approximation methods is regression on the basis of Gaussian processes [25] (in engineering applications the name kriging is usually used, see [14]). In the framework of this approach it is assumed that approximable function f can be represented as a linear combination of some known functions (for example, linear functions) and a realization of a Gaussian process with zero mean and covariance function from some parametric family. Approximation construction then reduces to estimation of covariance function parameters using a maximum likelihood method, and the coefficients of linear combination can be explicitly obtained using a least-squares method. Regression based on Gaussian processes is a flexible model with a rather small number of tunable parameters, and it provides accurate recovery of nonlinear dependences even in the case of small sample size. Moreover, due to the probabilistic nature of the model, Gaussian process-based regression provides not only approximation $\hat{f}$ (defined by the posterior mean), but also an accuracy evaluation capability (defined by the posterior variance of the process).

However, this approach has shortcomings. When tuning parameters of the covariance function using the maximum likelihood principle, it is necessary to invert the covariance matrix of size $N_{\text{learn}} \times N_{\text{learn}}$ during each iteration of the tuning process, and the complexity of each inversion is $O(N_{\text{learn}}^3)$. Thus, this operation requires a lot of computational resources (CPU time and memory) for big sample sizes. For $N_{\text{learn}} \gg 1{,}000$ such operations cannot be done within a reasonable time on a modern PC, which restricts the application of kriging if the sample size is big. There is a solution for this problem, based on an approximation of the covariance matrix [10, 25] with computational complexity $O(N_{\text{learn}} m^2)$, where m is the size of some subsample. A sufficiently big part of the initial sample is taken as the subsample, which allows us to widen the range of applicability of Gaussian process regression up to sample sizes equal to 10^4. Also, the use of a covariance matrix approximation instead of its true value decreases the approximation accuracy.

The method based on K nearest neighbors suffers from the *curse of dimensionality*: in the high-dimensional case the notion of vicinity degrades, and the distance to the nearest points becomes comparable to the distance to the faraway points, which decreases the approximation quality.

Another very popular approximation method is based on artificial neural networks (ANNs) [15, 16], including their special subclass: radial basis functions (RBFs). The use of ANNs is largely explained on the basis of theoretical results

(see, for example, [24]), which state that the ANN can provide approximation of a wide function class with any predefined accuracy. However, in practice this accuracy is not usually attained, since the theoretical results do not provide a practical method on how to construct an ANN (select the structure of the ANN, *tune* its parameters) for the given data sample S_{learn}. Nevertheless, an ANN is a flexible model which can be easily *extended* (by increasing the number of layers and/or their sizes) if the sample size grows. This flexibility also has drawbacks: ANN-based models have a lot of parameters (especially when there are many hidden layers), which prevents us from constructing robust/accurate approximations when only samples with small sizes are available. Moreover, model selection for an ANN is a very heuristic process.

The standard approach for *tuning* ANN parameters is an error backpropagation algorithm. Other methods exist for the tuning of parameters, which have faster convergence, for example, second-order methods, methods with adaptive learning rate, etc. However, these methods do not take into account the specific structure of the ANN, and in fact can be considered as applications of standard numerical optimization methods for tuning parameters in nonlinear regression. Also, the constructed approximation significantly depends on the (random) initialization of ANN parameters and (random) split of the learning sample into training and validation subsamples. So, ANN-based methods are rather flexible, but not very reliable for approximation construction.

For applications it is enough to consider only a two-layer perceptron as an ANN. Usually the hidden layer is composed either from sigmoid-type or RBF-type functions. However, in fact functions of different types can be mixed in one hidden layer, including, e.g., sigmoids, RBFs, and wavelets. A generalization of the two-layer perceptron can then be obtained with representation like that of Eq. (10). We call this kind of generalization *approximation by linear decomposition in nonlinear functions from a parametric dictionary* and we use the corresponding model as a base model in the HDA algorithm.

A typical algorithm for ANN learning can be decomposed into the following steps:

- Selection of the ANN structure,
- Random initialization of parameters,
- Random splitting of the learning sample into training and validation subsamples,
- Tuning of parameters using some gradient method for minimizing the mean-square error on the training subsample until the error on the validation subsample begins to increase.

However, such algorithms are mainly tailored for ANNs with general structure and do not take into account any specific structure of the model like (10). Therefore, in the HDA framework for all of these steps we have developed specific algorithms, which provide better results compared to the typical algorithms realized in commercial software. This allows us to obtain a good approximation accuracy; see Sect. 4.3.4 for more details.

4.3.4 Results of HDA Comparison with Conventional Approximation Methods

There are two possible ways to estimate the quality of the proposed approximation method. The first way is to check compliance with the requirements on accuracy, which are necessary for successful SM application. These requirements can be obtained only from the engineer and vary greatly for different problems. The second, more universal, method is to compare the quality of the proposed approach with conventional methods for approximation. We will consider realizations of such conventional methods in the software toolkits MatLab [17], modeFrontier [18], which are widely used in many industrial companies. In each of these software toolkits there are several methods for approximation of multidimensional dependency, including ANNs, regression based on Gaussian processes, etc.

A priori two main shortcomings of these toolkits can be pointed out. First, in order to select a particular approximation method for solving a given problem, an engineer must have either some knowledge of machine learning or must perform numerical experiments using all available approximation methods. Otherwise, one might select an approximation method which does not provide the best approximation quality. Moreover, many implemented approximation methods have strong limitations on the input dimension p and sample size N_{learn}, which in general decreases the approximation quality.

Let us consider the results on some indicative problems, covering a wide range of sample sizes and input dimensions (these results were obtained in 2010 during work on the PhD thesis):

- Airplane fuselage composite structure design (data from the Airbus in-house stability tools approximation problem considered here; see also [6] for details),
- Radiator characteristics modeling using dipoles,
- Fuel consumption of an airplane engine.

The accuracy of approximation is estimated using the square root of an error (3) on the test set, divided by the range of the corresponding output.

Remark 1 For the data from the airplane fuselage composite structure design problem, all computational experiments were made on the so-called accuracy domain $\tilde{\mathbf{X}} = \{\mathbf{x} : RF(\mathbf{x}) \in (1-\varepsilon, 1+\varepsilon)\}$, $\varepsilon = 0.2$. This means that the training (1) and test samples had the forms $\tilde{S}_{\text{learn}} = \{(\mathbf{x}, y) \in S_{\text{learn}} : \mathbf{x} \in \tilde{\mathbf{X}}\}$ and $\tilde{S}_{\text{test}} = \{(\mathbf{x}, y) \in S_{\text{test}} : \mathbf{x} \in \tilde{\mathbf{X}}\}$, respectively.

The results of the approximation accuracy comparison are given in Table 1.

Further, let us consider a significantly bigger set of problems (including three problems considered above), based on data from physical experiments or from simulation experiments with some physical model. The characteristics of the corresponding learning samples are rather diverse:

- Input dimension p varies from 2 to 435; the output dimension varies from 1 to 100.

Table 1 Relative approximation errors

Problem name		Composite structure	Dipole	Fuel cons.
Dimension p		16	7	3
Sample size N_{learn}		50,000	5,000	351
MACROS	HDA	**0.0120**	**0.0019**	**0.0044**
MatLab	Linear reg.	0.0806	0.1137	0.2075
	Quadratic reg.	0.0525	0.0971	0.1180
	RBF	**0.0454**	0.0246	0.0172
modeFrontier	K-nearest	0.0753	0.0811	0.0777
	Anisotropic kriging	0.0804	**0.0071**	**0.0126**
	Kriging	**0.0496**	0.0359	0.1845
	RBF	**0.0496**	0.0177	0.0375
	ANN	0.0503	**0.0037**	**0.0147**
	Evolutionary design	0.2242	0.1570	0.0266
	Gaussian processes	0.0898	0.0612	0.0211

- The size of the learning sample is up to 65,000 points.
- The number of considered problems is 30.

For each of these problems we constructed approximations using all available methods. For convenient representation of the comparison results on the full set of problems we used the visualization method described in [13]. This approach provides a graphical comparison of approximations accuracies for several considered methods and a large number of test problems. Let us briefly describe this visualization method using the following notation:

- $M_k, k = 1, \ldots, K$ are considered approximation methods,
- $P_l, l = 1, \ldots, L$ are considered test problems,
- $e(M_k, P_l)$ is a root mean square approximation error for the approximator, constructed using method M_k on the problem P_l,
- $\widetilde{e}(P_l) = \min_k e(M_k, P_l)$ is a minimal (among all available approximation methods) error for the problem P_l.

For any method M_k and scaling factor $a \geq 1$ let us define the following quantity:

$$P_k(a) = \frac{\#\{l : e(M_k, P_l) \leq a \cdot \widetilde{e}(P_l)\}}{L}.$$

In fact, the quantity $P_k(a)$ shows on which part of problems approximation errors of the method M_k are not a times bigger than minimal (among methods) approximation errors for the corresponding problems. In particular, $P_k(1)$ is equal to the fraction of problems for which the method M_k provides the smallest approximation error.

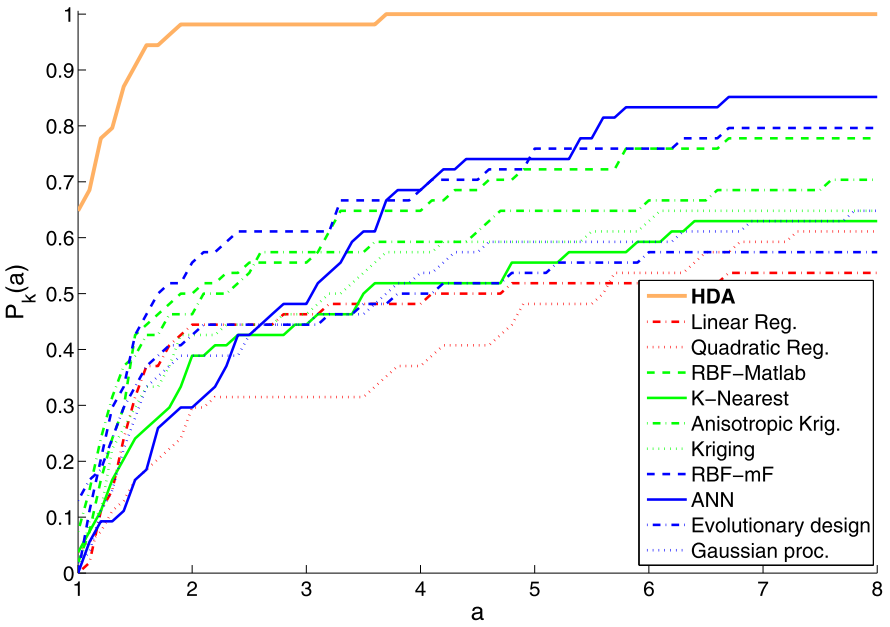


Fig. 7 Results of comparison

According to the obtained results (see Fig. 7), on 65 % of problems the HDA algorithm has the smallest error of approximation. Moreover, profile $P(a)$ of the HDA algorithm converges to 1 rather fast, which also shows its robustness. Therefore, this algorithm provides significantly better accuracy compared to conventional methods, realized in commercial products. Similar results were obtained during the internal comparison of GT Approx in Airbus with in-house software realizing different approximation methods.

4.3.5 Comparison of HDA with IMAGE

A comparison of IMAGE and HDA with conventional methods on the problem of Airbus in-house stability tools approximation showed that these methods had a significantly bigger accuracy than that of the conventional ones (for IMAGE see [4] and for HDA see [6], Sect. 4.3.4). Therefore, the main goal of this section is to compare the accuracies of IMAGE and HDA.

4.3.6 Setup of Experiments

The setup of the experiments can be described as follows:

- Input dimension $p = 20$, input vector defines geometries, material properties, etc.

- Typical learning sample size, used for construction of an approximation, is $N_{\text{learn}} \sim 200\,000$.
- Output characteristics consist of various reserve factors (RF for stringer local buckling, RF for skin local buckling, etc.) and strain values at different locations, also considering bending. The number of output characteristics to approximate is equal to 25.

An RF indicates whether the structure is feasible with respect to a given failure mode. If the RF is greater than one, the structure is feasible. If the RF is less than one it is not feasible. Therefore, when modeling the dependency of an RF on a vector of design variables $\mathbf{x}$, the highest possible accuracy should be provided for values of RF close to one. Thus for each output characteristic two domains were identified, $\mathbf{X}_1 = \{\mathbf{x} \in \mathbf{X} : f(\mathbf{x}) \in [0.5, 1.5]\}$ and $\mathbf{X}_2 = \{\mathbf{x} \in \mathbf{X} : f(\mathbf{x}) \in [0, 3]\}$, and for each of these domains an approximator was constructed using the corresponding train sample. Let us denote these approximators as $\hat{f}_1$ and $\hat{f}_2$, respectively.

As accuracy indicators we used $Emean$ (mean absolute error), $Eq95$ (95 % quantile absolute error), and $Eq99$ (99 % quantile absolute error). Accuracy indicators were estimated using separate test samples for sets $\mathbf{X}_{1,1} = \{\mathbf{x} \in \mathbf{X} : f(\mathbf{x}) \in [0.8, 1.2]\}$ and $\mathbf{X}_{1,2} = \{\mathbf{x} \in \mathbf{X} : f(\mathbf{x}) \in [1.2, 1.5]\}$ in the case of approximator $\hat{f}_1$ and for sets $\mathbf{X}_{2,1} = \{\mathbf{x} \in \mathbf{X} : f(\mathbf{x}) \in [0, 1.5]\}$ and $\mathbf{X}_{2,2} = \{\mathbf{x} \in \mathbf{X} : f(\mathbf{x}) \in [1.5, 3]\}$ in the case of approximator $\hat{f}_2$.

4.3.7 Obtained Results

Due to lack of space, detailed results are given only for two RFs: RF for stringer local buckling (RF STR, see Table 2) and RF for first skin buckling mode (RF PND GEN, see Table 3). We can see that HDA provides significantly higher accuracy.

Let us consider the fraction of cases (out of 50) for which the considered approximation method has the smallest error (of considered type); see the results in Table 4. We can see that for most cases HDA achieves the smallest approximation errors.

Let us quantify to what extent on average the accuracy of HDA is higher than the accuracy of IMAGE. We denote by $E(M)$ an approximation error of some particular type, obtained by a method M on some test problem, $\Delta(M_1, M_2) = (E(M_1) - E(M_2))/E(M_1) \cdot 100$ % for some two methods M_1 and M_2. Then the average gain (*AvgGain*) of using M_2 instead of M_1 is equal to the average of positive values of $\Delta(M_1, M_2)$ over all considered problems. Analogously, the average loss (*AvgLoss*) of using M_2 instead of M_1 is equal to the average of negative values of $\Delta(M_1, M_2)$ over all considered problems. The obtained results for IMAGE (M_1) and HDA (M_2) are given in Table 5. We can see that HDA is significantly more accurate.

Table 2 Accuracy Indicators for RF STR

	Error	IMAGE				HDA
		30 experts	40 experts	45 experts	50 experts	
$\hat{f}_1$ for RF STR						
$\mathbf{x} \in \mathbf{X}_{1,1}$	*Emean*	0.0579	0.0172	0.0279	0.0295	0.0046
	*Eq*95	0.2403	0.1115	0.1528	0.1665	0.0145
	*Eq*99	0.3435	0.1873	0.2453	0.2609	0.0362
$\mathbf{x} \in \mathbf{X}_{1,2}$	*Emean*	0.0982	0.0323	0.0462	0.0492	0.0078
	*Eq*95	0.3639	0.1427	0.1952	0.2122	0.0219
	*Eq*99	0.5378	0.3024	0.3588	0.3782	0.0893
$\hat{f}_2$ for RF STR						
$\mathbf{x} \in \mathbf{X}_{2,1}$	*Emean*	0.1105	0.1039	0.0990	0.0961	0.0056
	*Eq*95	0.3829	0.3636	0.3530	0.3451	0.0191
	*Eq*99	0.5785	0.5526	0.5350	0.5479	0.0599
$\mathbf{x} \in \mathbf{X}_{2,2}$	*Emean*	0.2395	0.2294	0.2254	0.2208	0.0186
	*Eq*95	0.7099	0.7079	0.6857	0.6734	0.0525
	*Eq*99	1.2116	1.2142	1.2116	1.1529	0.1807

Table 3 Accuracy indicators for RF PND GEN

	Error	IMAGE				HDA
		30 experts	40 experts	45 experts	50 experts	
$\hat{f}_1$ for RF PND GEN						
$\mathbf{x} \in \mathbf{X}_{1,1}$	*Emean*	0.0526	0.0575	0.0577	0.0596	0.0399
	*Eq*95	0.2092	0.2212	0.2134	0.2196	0.1136
	*Eq*99	0.3060	0.3153	0.2931	0.2991	0.1784
$\mathbf{x} \in \mathbf{X}_{1,2}$	*Emean*	0.0829	0.0849	0.0841	0.0854	0.0579
	*Eq*95	0.2656	0.2652	0.2528	0.2650	0.1753
	*Eq*99	0.4085	0.3960	0.3900	0.3981	0.2926
$\hat{f}_2$ for RF PND GEN						
$\mathbf{x} \in \mathbf{X}_{2,1}$	*Emean*	0.2679	0.2555	0.2500	0.2498	0.0530
	*Eq*95	0.6374	0.5808	0.5983	0.6011	0.1623
	*Eq*99	0.8298	0.7671	0.7793	0.7967	0.2655
$\mathbf{x} \in \mathbf{X}_{2,2}$	*Emean*	0.2731	0.2597	0.2538	0.2470	0.1039
	*Eq*95	0.7476	0.7033	0.6893	0.6745	0.3238
	*Eq*99	1.0702	1.0016	0.9907	0.9784	0.5705

Table 4 Fraction of cases

Error	HDA	IMAGE
$Emean$	0.853	0.147
$Eq95$	0.941	0.059
$Eq99$	1	0.000

Table 5 Average gain and loss

Error		HDA vs IMAGE
$Emean$	$AvgGain\,(\%)$	56.20
	$AvgLoss\,(\%)$	−8.02
$Eq95$	$AvgGain\,(\%)$	54.27
	$AvgLoss\,(\%)$	−1.65
$Emean$	$AvgGain\,(\%)$	48.56
	$AvgLoss\,(\%)$	0.00

5 Results of Optimization Based on Skill Tool and MSM

In this section some results of using SMs in the pre-sizing optimization tool COMBOX for composite structures are presented. The classical skill tool called PS3 (Plane Super-Stiffener Sizing) was replaced by the MACROS Surrogate Model (MSM). The objective of this study was to check the impact of this replacement on the accuracy, on the convergence of the optimization process, and on the run time.

5.1 Optimization Runs

In this study the optimization was performed on the wing lower and upper covers, as described in Fig. 8. Two test cases for the optimization study were considered corresponding to two starting points:

- The first one is close to an optimal design, obtained by using only the PS3 skill tool.
- The second one is a heavy one, where all design variables are set to their upper bound. This last run illustrates the behavior using SMs for a complete optimization run.

For each test case, the first run was done with MSM (until convergence), and then update and restart with PS3 was performed. Due to limited space only some representative results are shown.

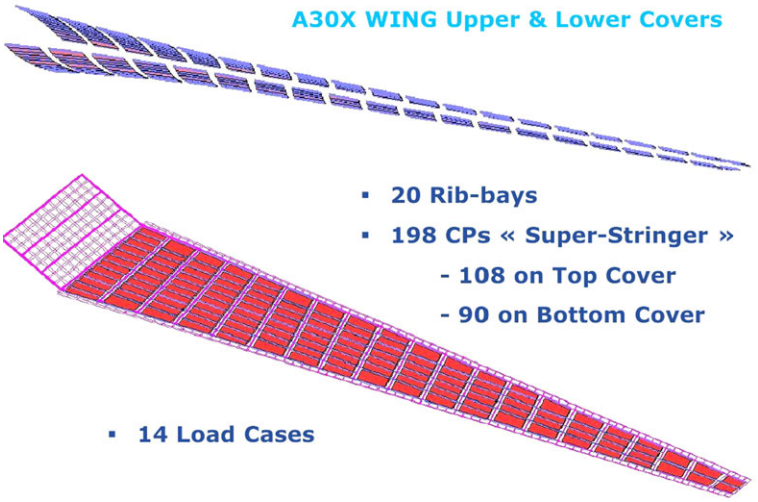


Fig. 8 A30X wing stress model

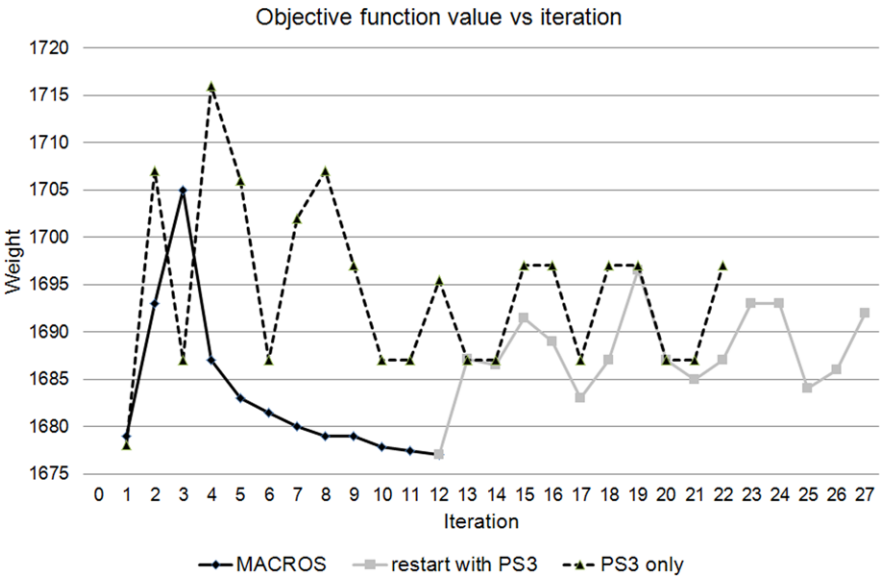


Fig. 9 Evolution of the objective function for initial starting point

5.1.1 Optimization Runs: Initial Starting Point

The comparison with the *pure* PS3 run is presented in Fig. 9 (solid black-gray curve corresponds to MSM and subsequent update with PS3; dashed curve corresponds to the run with only PS3).

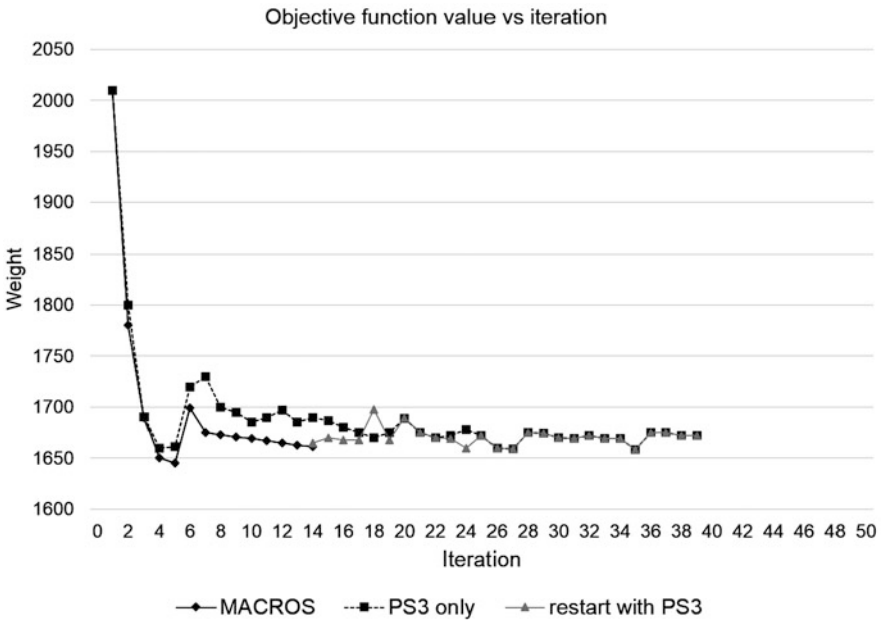


Fig. 10 Evolution of the objective function for heavy starting point

We can observe a more chaotic evolution with PS3 which is not only due to proximity to the optimal solution, but is also linked to some discontinuities in the RFs calculated with PS3.

5.1.2 Optimization Runs: Heavy Starting Point

The evolution of the objective value is given in Fig. 10 (black solid curve corresponds to MSM, gray solid curve corresponds to restart from MSM with PS3, dashed curve corresponds to PS3 only). We observe a smooth evolution with both MSM (SM is a continuous function) and PS3 (the behavior of PS3 is smoother than in the previous run, since the starting point is not as close to the optimal design), but the evolution is nevertheless smoother with MSM.

The evolution of the numbers of violated constraints is given in Fig. 11 and saturated constraints is given in Fig. 12. The peak at the beginning of the left plot appears due to the strategy of active constraints. We can observe an increase of the number of violated constraints when updating the model (due to the switch of skill tool) and after a quick decrease of it. A decrease of the number of saturated constraints at iteration six occurs due to the reactualization of the active constraints and the identification of new violated constraints; a decrease of the number of saturated constraints from MSM to restart with PS3 occurs due to the switch of the skill tool. The restart with PS3 presents less saturated constraints than MSM. This proves a

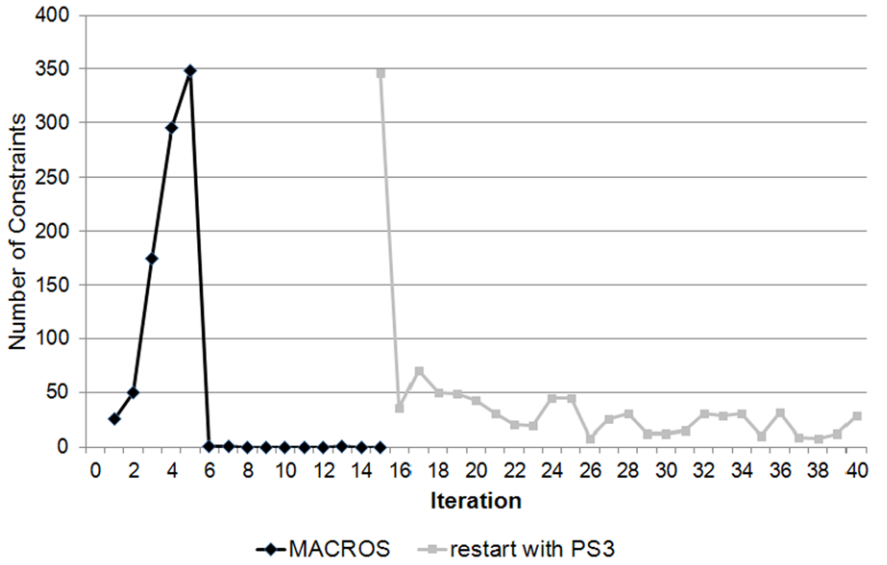


Fig. 11 Evolution of the number of violated constraints

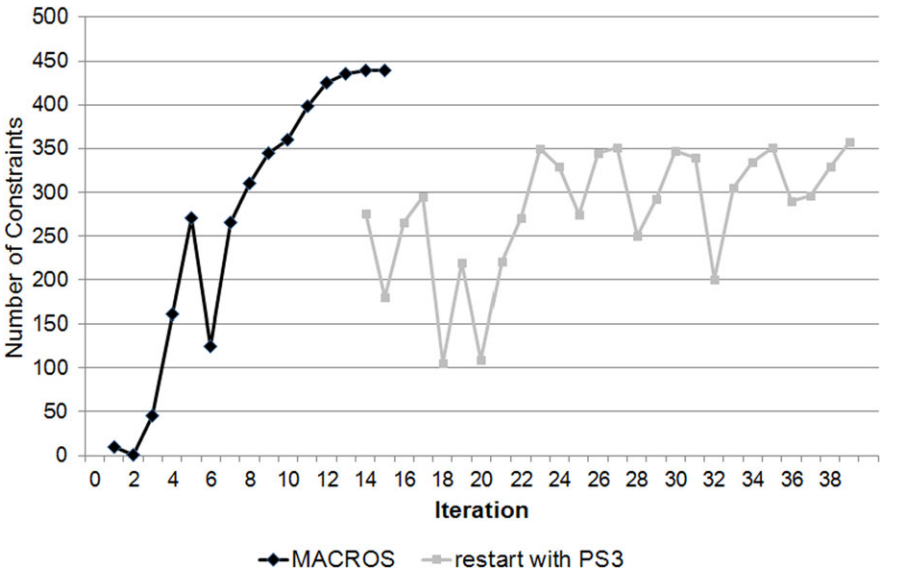
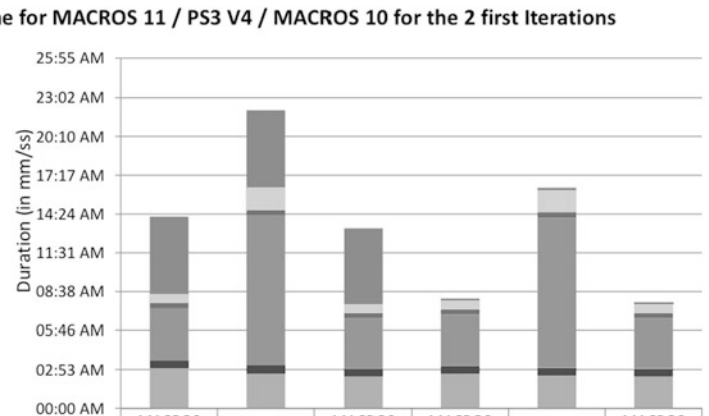


Fig. 12 Evolution of the number of saturated constraints

better quality of the optimum found with MSM, which is linked to the smoothness and mathematical differentiability of MSM, in contrast to PS3.



	MACROS 14.1	PS3V4	MACROS 10	MACROS 14.1	PS3V4	MACROS 10
	ITERATION 1			ITERATION 2		
NASTRAN SOL 200	00:05:49	00:05:45	00:05:38	00:00:09	00:00:11	00:00:08
MACROS or PS3 Nominal	00:00:37	00:01:40	00:00:38	00:00:38	00:01:39	00:00:37
Inter-regional Constraints Nominal	00:00:25	00:00:25	00:00:24	00:00:24	00:00:25	00:00:25
MACROS or PS3 Sensitivities	00:03:50	00:11:06	00:03:44	00:03:48	00:11:06	00:03:42
Inter-regional Constraints Sensitivities	00:00:29	00:00:28	00:00:29	00:00:28	00:00:28	00:00:28
Additional Time	00:03:03	00:02:43	00:02:27	00:02:40	00:02:32	00:02:29

Fig. 13 Run Time (hh/mm/ss) for MACROS 14.1/PS3 V4/MACROS 10 on iterations 1 and 2

5.2 Execution Time

The objective of this study was to compare the run time of the MSM and PS3 skill tools for one iteration of the optimization process. The study was done on test case 1 (initial point), during the first two iterations. The sequence of computations for one iteration is the following:

- NASTRAN SOL200: for internal load update and sensitivity analysis,
- Computation of the RFs corresponding to the current design with the PS3 stress process or the MSM,
- Computation of the interregional constraints corresponding to the current design,
- Computation (by using PS3 or MSM) of the sensitivities for all the RFs,
- Computation of the sensitivities for the interregional constraints.

We observed an overall gain factor of at least 2.5 by using MSM instead of PS3 (because of the acceleration of the steps *MACROS or PS3 Nominal* and *MACROS or PS3 Sensitivities*, see Fig. 13). Since MSM provides very fast analytical computation of sensitivities, further significant speedup can be obtained by using analytical sensitivity computations instead of numerical ones. The speedup factor is less than expected, because much system time is spent in the management of jobs via IBM Platform LSF (Load Sharing Facility, a workload management platform for demanding, distributed HPC environments, see [22]).

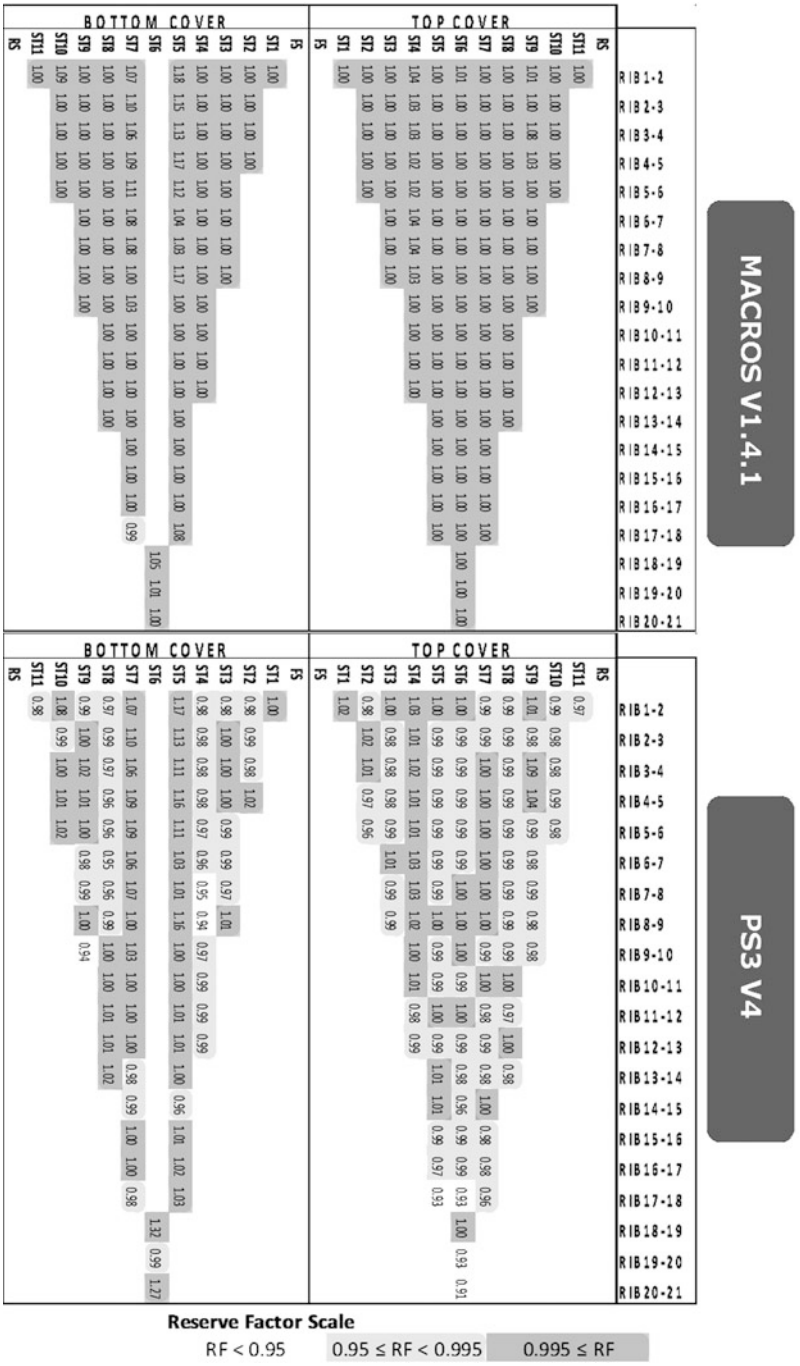


Fig. 14 Check with PS3 of optimum results found with MACROS SM

5.3 Check of Reserve Factors

A check of RFs was performed with the optimum based on MSM. The results are presented in Fig. 14. These results show a satisfactory accuracy for a pre-sizing result, according to Airbus experts, considering that a pre-sizing is always to be re-engineered including, e.g., manufacturing constraints. Work is ongoing to further improve the accuracy.

6 Conclusion

A constructed MACROS Surrogate Model (MSM) was embedded into the pre-sizing optimization process of A350XWB boxes realized in the pre-sizing tool COMBOX for checking the validity of the approximation and its usage as a constraint in an optimization process. An analysis/comparison of optimization results based on a skill tool and optimization results based on the constructed MSM was performed and showed that MSM:

- gives a high accuracy of approximation (see also [6]),
- allows one to obtain smoother convergence in fewer iterations with a smoother distribution of thickness/stringer dimensions and a small violation of constraints, which could be easily repaired at the detailed design phase,
- provides a *reduction of structure optimization computational time* from several days to a few hours.

Acknowledgements E. Burnaev, M. Belyaev, and P. Prikhodko were partially supported by the Laboratory for Structural Methods of Data Analysis in Predictive Modeling, MIPT, RF government grant, ag. 11.G34.31.0073.

References

1. Belyaev, M., Lyubin, A.: Some features of optimization problem arising in construction of multidimensional approximation. In: Proceedings of the Conference for Young Scientists and Engineers “Information Technology and Systems—2011”, Gelendzhik, Russia, pp. 415–422 (2011)
2. Belyaev, M.G., Burnaev, E.V., Lyubin, A.D.: Parametric dictionary selection for approximation of multidimensional dependency. In: Proceedings of All-Russian Conference “Mathematical Methods of Pattern Recognition (MMPR-15)”, Petrozavodsk, 11–17 September 2011, pp. 146–150 (2011)
3. Belyaev, M., Burnaev, E., Yerofeyev, P., Prikhodko, P.: Comparative performance of nonlinear regression initialization methods. In: Proceedings of the Conference for Young Scientists and Engineers “Information Technology and Systems—2011”, Gelendzhik, Russia, pp. 315–320 (2011)
4. Bettebghor, D., Bartoli, N., Grihon, S., Morlier, J., Samuelides, M.: Surrogate modelling approximation using a mixture of experts based on EM joint estimation. Struct. Multidiscip. Optim. **43**(2), 243–259 (2011)

5. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Berlin (2006)
6. Burnaev, E.V., Grihon, S.: Construction of the metamodels in support of stiffened panel optimization. In: Proceedings of the International Conference on Mathematical Methods in Reliability, Moscow, Russia, pp. 124–128 (2009)
7. Burnaev, E., Prikhodko, P.: Theoretical properties of procedure for construction of regression ensemble based on bagging and boosting. In: Proceedings of the Conference for Young Scientists and Engineers "Information Technology and Systems—2011", Gelendzhik, Russia, pp. 438–443 (2011)
8. Burnaev, E., Prikhodko, P.: Approaches to dimensionality estimation in the effective dimensionality reduction based on Gaussian processes. In: Information Technologies and Systems 2012, Petrozavodsk, August 19–25. Springer, Berlin (2012)
9. Burnaev, E.V., Belyaev, M.G., Prihodko, P.V.: About hybrid algorithm for tuning of parameters in approximation based on linear expansions in parametric functions. In: Proceedings of the 8th International Conference "Intellectualization of Data Processing" (IDP-2010), Cyprus, 17–24 October 2011, pp. 200–203 (2011)
10. Burnaev, E.V., Zaytsev, A.A., Yanovich, Yu.A.: Consolidation of data with different fidelity on basis of Gaussian processes. In: Proceedings of the Conference "Information Technologies and Systems", pp. 60–65 (2012)
11. Chiles, J.-P., Delfiner, P.: Geostatistics. Modeling Spatial Uncertainty. Wiley, New York (1999)
12. DATADVANCE: www.datadvance.net
13. Dolan, E., More, J.: Benchmarking optimization software with performance profiles. Math. Program., Ser. A **91**, 201–213 (2002)
14. Forrester, A., Sobester, A., Keane, A.: Engineering Design via Surrogate Modeling. A Practical Guide. Wiley, New York (2008)
15. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, Berlin (2008)
16. Haykin, S.S.: Neural Networks: A Comprehensive Foundation. Williams, Baltimore (2006)
17. http://en.wikipedia.org/wiki/Matlab
18. http://en.wikipedia.org/wiki/ModeFRONTIER
19. http://en.wikipedia.org/wiki/Nastran
20. http://www.lmsintl.com/simulation/caesam
21. http://www.lmsintl.com/samtech-boss-quatro
22. http://www-03.ibm.com/systems/technicalcomputing/platformcomputing/products/lsf/index.html
23. Kuleshov, A.P., Bernstein, A.V., Burnaev, E.V.: Adaptive models of complex systems based on data handling. In: Proceedings of the Third International Conference on Inductive Modelling, Kyiv, Ukraine, pp. 64–71 (2010)
24. Pinkus, A.: Approximation theory of the MLP model in neural networks. Acta Numer. **8**, 143–195 (1999)
25. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2006)
26. Wang, G., Shan, S.: Review of metamodeling techniques in support of engineering design optimization. J. Mech. Des. **129**(3), 370–381 (2007)

Engineering Optimization and Industrial Applications

Xin-She Yang

Abstract Design optimization is important in engineering and industrial applications. It is usually very challenging to find optimum designs, which require both efficient optimization algorithms and high-quality simulators that are often time-consuming. To some extent, an optimization process is equivalent to a self-organizing system, and the organized states are the optima that are to be searched for. In this chapter, we discuss both optimization and self-organization in a unified framework, and we use three metaheuristic algorithms, the firefly algorithm, the bat algorithm and cuckoo search, as examples to see how this self-organized process works. We then present a set of nine design problems in engineering and industry. We also discuss the challenging issues that need to be addressed in the near future.

Keywords Bat algorithm · Cuckoo search · Firefly algorithm · Optimization · Metaheuristic · Self-organizaion

1 Introduction

Optimization is ubiquitous in many applications in engineering and industry. In essence, optimization is a process of searching for the optimal solutions to a particular problem of interest, and this search process can be carried out using multiple agents which essentially form a system of evolving agents. This system can evolve by iterations according to a set of rules or mathematical equations. Consequently, such systems will show some emergent characteristics, leading to self-organizing states which correspond to some optima in the search space. Once the self-organized states are reached, we say the system has converged. Therefore, the design of an efficient optimization algorithm is equivalent to mimicking the evolution of a self-organizing system.

In almost all applications in engineering and industry, we are always trying to optimize something—whether to minimize the cost and energy consumption, or to

X.-S. Yang (✉)
School of Science and Technology, Middlesex University, London NW4 4BT, UK
e-mail: X.Yang@mdx.ac.uk

S. Koziel, L. Leifsson (eds.), *Surrogate-Based Modeling and Optimization*,
DOI 10.1007/978-1-4614-7551-4_16,

maximize the profit, output, performance and efficiency. In reality, resources, time and money are always limited; consequently, optimization is far more important in practice [2, 13, 30, 32, 36, 39, 41].

It is worth pointing out that computational efforts are a main issue in many optimization problems in engineering and industry, because the most time-consuming part of the optimization process is the evaluations of objective functions [17, 39]. The use of the most efficient algorithms is just one way of tackling the problem, while an alternative is to use surrogate-based models which can often be more efficient if the number of evaluating high-fidelity models is significantly reduced [18–20]. Such surrogate-based optimization using a combination of low-fidelity and high-fidelity models can be a powerful tool for many real-world applications. This book contains many examples of surrogate-based modelling and optimization. In this chapter, our focus is mainly on the introduction of some widely used new algorithms and a well-chosen set of design benchmarks.

2 Optimization Algorithms and Self-organization

2.1 Self-organizing Systems

Self-organization exists in many systems, from physical and chemical to biological and artificial systems. Emergent phenomena such as Rayleigh–Bénard convection, Turing pattern formation [26] and organisms and thunderstorms [15] can all be called self-organizing. Though there is no universal theory for self-organizing processes, some aspects of self-organization can be partly understood using theories based on nonlinear dynamical systems, far-from-equilibrium [23] multiple interacting agents, and closed systems under unchanging laws [3]. As pointed out by cyberneticist and mathematician Ross Ashby, every isolated determinate dynamic system, obeying unchanging laws, will ultimately develop some sort of 'organisms' that are adapted to their 'environments' [3].

For simple systems, going to equilibrium is trivial, but, for a complex system, if its size is so large that its equilibrium states are just a fraction of the vast number of possible states, and if the system is allowed to evolve long enough, some self-organized structures may emerge. Changes in environment can apply pressure on the system to re-organize and adapt to such changes. If the systems have sufficient perturbations or noise, often working at the edge of the chaos, some spontaneous formation of structures will emerge as the systems move, far from equilibrium, and select some states, thus reducing the uncertainty or entropy.

The state set S of a complex system such as a machine may change from initial states $S(\psi)$ to other states $S(\phi)$, subject to the change of a parameter set $\alpha(t)$ which can be time-dependent. That is,

$$S(\psi) \xrightarrow{\alpha(t)} S(\phi), \tag{1}$$

where $\alpha(t)$ must come from external conditions such as the heat flow in Rayleigh–Bénard convection, not from the states S themselves. Obviously, $S + \alpha(t)$ can be

considered as a larger, closed system [3]. In this sense, self-organization is equivalent to a mapping from some high-entropy states to low-entropy states.

An optimization algorithm can be viewed as a complex, dynamical system. If we can consider the convergence process as a self-organizing process, then there are strong similarities and links between self-organizing systems and optimization algorithms.

2.2 Algorithms for Self-organization

Mathematically speaking, an algorithm is a procedure to generate outputs for given inputs. From the optimization point of view, an optimization algorithm generates a new solution $\boldsymbol{x}^{t+1}$ to a given problem from a known solution $\boldsymbol{x}^t$ at iteration or time t. That is

$$\boldsymbol{x}^{t+1} = \boldsymbol{A}\big(\boldsymbol{x}^t, \boldsymbol{p}(t)\big), \tag{2}$$

where $\boldsymbol{A}$ is a nonlinear mapping from a given solution, or d-dimensional vector, $\boldsymbol{x}^t$ to a new solution vector $\boldsymbol{x}^{t+1}$. The algorithm $\boldsymbol{A}$ has k algorithm-dependent parameters $\boldsymbol{p}(t) = (p_1, \ldots, p_k)$ which can be time-dependent and can thus be tuned if necessary.

To find the optimal solution $\boldsymbol{x}_*$ to a given optimization problem S with an often infinite number of states is to select some desired states ϕ from all states ψ, according to some predefined criterion D. We have

$$S(\psi) \xrightarrow{\boldsymbol{A}(t)} S\big(\phi(x_*)\big), \tag{3}$$

where the final converged state ϕ corresponds to an optimal solution $\boldsymbol{x}_*$ to the problem of interest. The selection of the system states in the design space is carried out by running the optimization algorithm $\boldsymbol{A}$. The behaviour of the algorithm is controlled by $\boldsymbol{p}$, the initial solution $\boldsymbol{x}^{t=0}$ and the stopping criterion D. We can view the combined $S + \boldsymbol{A}(t)$ as a complex system with a self-organizing capability.

The change of states or solutions of the problem of interest is controlled by the algorithm $\boldsymbol{A}$. In many classical algorithms such as hill-climbing, gradient information is often used to select states, say, the minimum value of the landscape, and the stopping criterion can be a given tolerance or accuracy, or zero gradient etc. Alternatively, an algorithm can act like a tool to tune a complex system. If an algorithm does not use any state information of the problem, then it is more likely to be versatile to deal with many types of problems. However, such black-box approaches can also imply that the algorithm may not be as efficient as it could be for a given type of problem. For example, if the optimization problem is convex, algorithms that use such convexity information will be more efficient than those that do not use such information. In order to select states/solutions efficiently, the information from the search process should be used to enhance the search process. In many cases, such information is often fed into the selection mechanism of an algorithm. By far

the most widely used selection mechanism is to identify and keep the best solution found so far. That is, some form of 'survival of the fittest' is used.

Optimization algorithms can be very diverse. There are dozens of widely used algorithms. The main characteristics of different algorithms will only depend on the actual, often highly nonlinear or implicit, forms of $\boldsymbol{A}(t)$ and their parameters $\boldsymbol{p}(t)$.

In many situations concerning optimization, the generation and verification of the new solutions can often involve computationally expensive computer simulations or even measurements of the physical system. In such cases, the expensive model of the system under consideration is often replaced by its cheaper representation, called a surrogate model, and the algorithm A uses that model to produce a new solution. The parameters $p(t)$ may then include variables that are used to align the surrogate with the expensive model to make it a reliable representation of the latter.

3 Three New Algorithms

In this chapter, we illustrate the concept of a self-organizing optimization algorithm using a specific class of algorithms called metaheuristics. Metaheuristics have some important characteristics that uses stochastic components to enable an algorithm to escape the possibility of being trapped in a local optimum. This often makes the search process more ergodic, and thus such algorithms can generate high-quality solutions over the search space during iterations, which may ultimately converge towards the true optimality of the problem of interest.

There are well over two dozen metaheuristic algorithms now in use for optimization [16, 30, 34]. All metaheuristic algorithms have to balance exploration and exploitation during the search process by using some sort of algorithm-dependent parameter setting. From the viewpoint of a self-organizing system, parameter settings will affect the way and routes by which the optimization process converges to an organized state. Here we analyse three relatively new nature-inspired algorithms and see the ways in which they can quickly converge towards optimality.

3.1 Firefly Algorithm

The first algorithm to be discussed is the firefly algorithm, which is essentially a dynamical system. The firefly algorithm (FA), first developed by Xin-She Yang in 2008 [29, 30], was based on the flashing patterns and behaviour of fireflies. In essence, FA uses the following three idealized rules:

- Fireflies are unisex, so one firefly can be attracted to any other one.
- The attractiveness is proportional to the brightness and they both decrease as their distance increases. Thus for any two flashing fireflies, the less brighter one will move towards the brighter one. If there is no brighter one than a particular firefly, it will move randomly.

- The brightness of a firefly is determined by the landscape of the objective function.

As a firefly's attractiveness is proportional to the light intensity seen by adjacent fireflies, we can now define the variation of attractiveness β with the distance r by

$$\beta = \beta_0 e^{-\gamma r^2}, \tag{4}$$

where β_0 is the attractiveness at $r = 0$.

The movement of a firefly i that is attracted to another more attractive (brighter) firefly j is determined by

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \beta_0 e^{-\gamma r_{ij}^2} \left(\boldsymbol{x}_j^t - \boldsymbol{x}_i^t\right) + \alpha \boldsymbol{\epsilon}_i^t, \tag{5}$$

where the second term is due to the attraction. The third term is randomization with α being the randomization parameter, and $\boldsymbol{\epsilon}_i^t$ is a vector of random numbers drawn from a Gaussian distribution or uniform distribution at time t. If $\beta_0 = 0$, it becomes a simple random walk. Furthermore, the randomization $\boldsymbol{\epsilon}_i^t$ can easily be extended to other distributions such as Lévy flights. A Lévy flight essentially provides a random walk whose random step length is drawn from a Lévy distribution

$$\text{Lévy} \sim u = t^{-\lambda} \quad (1 < \lambda \leq 3), \tag{6}$$

which has an infinite variance with an infinite mean. Here the steps essentially form a random walk process with a power-law step-length distribution with a heavy tail. Some of the new solutions should be generated by the Lévy walk around the best solution obtained so far, and this will speed up the local search. A demo version of firefly algorithm implementation, without Lévy flights, can be found at the Mathworks file exchange web site.[1] FA has attracted much attention recently [1, 11, 25].

A discrete version of FA can efficiently solve NP-hard scheduling problems [25], while a detailed analysis has demonstrated the efficiency of FA over a wide range of test problems, including multiobjective load dispatch problems [1]. Highly nonlinear and non-convex global optimization problems can be solved efficiently using FA [11, 42].

From the self-organization point of view, FA acts as a simple dynamic system with diverse characteristics that can automatically subdivide the entire population into subgroups, and each subgroup can swarm around a local mode. Among all the local modes, there is always a global optimum, and thus FA can find the global optimality and local optima simultaneously if the number of fireflies is sufficiently higher than the number of modes.

[1] http://www.mathworks.com/matlabcentral/fileexchange/29693-firefly-algorithm.

3.2 Cuckoo Search

Cuckoo search (CS) is one of the latest nature-inspired metaheuristic algorithms, developed in 2009 by Xin-She Yang and Suash Deb [34]. CS is based on the brood parasitism of some cuckoo species. In addition, this algorithm is enhanced by Lévy flights, rather than by simple isotropic random walks. Recent studies show that CS is potentially far more efficient than particle swarm optimization (PSO) and genetic algorithms [35].

Cuckoos are fascinating birds, not only because of the beautiful sounds they can make, but also because of their aggressive reproduction strategy. Some species such as the *Ani* and *Guira* cuckoos lay their eggs in communal nests, though they may remove others' eggs to increase the hatching probability of their own eggs. Quite a number of species engage the obligate brood parasitism by laying their eggs in the nests of other host birds (often other species).

For simplicity in describing CS, we now use the following three idealized rules:

- Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest.
- The best nests with high-quality eggs will be carried over to the next generations.
- The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability $p_a \in [0, 1]$. In this case, the host bird can either get rid of the egg, or simply abandon the nest and build a completely new nest.

As a further approximation, this last assumption can be approximated by a fraction p_a of the n host nests that are replaced by new nests (with new random solutions). For a maximization problem, the quality or fitness of a solution can simply be proportional to the value of the objective function. Other forms of fitness can be defined in a similar way to the fitness function in genetic algorithms.

This algorithm uses a balanced combination of a local random walk and a global explorative random walk, controlled by a switching parameter p_a. The local random walk can be written as

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \alpha s \otimes H(p_a - \epsilon) \otimes \left(\boldsymbol{x}_j^t - \boldsymbol{x}_k^t\right), \tag{7}$$

where $\boldsymbol{x}_j^t$ and $\boldsymbol{x}_k^t$ are two different solutions selected randomly by random permutation, $H(u)$ is a Heaviside function, ϵ is a random number drawn from a uniform distribution, and s is the step size. On the other hand, the global random walk is carried out by using Lévy flights

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \alpha L(s, \lambda), \tag{8}$$

where

$$L(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin(\pi \lambda / 2)}{\pi} \frac{1}{s^{1+\lambda}} \quad (s \gg s_0 > 0). \tag{9}$$

Here $\alpha > 0$ is the step size scaling factor, which should be related to the scales of the problem of interest. In most cases, we can use $\alpha = O(L/10)$, where L is the

characteristic scale of the problem of interest, while in some case $\alpha = O(L/100)$ can be more effective and avoid flying too far.

The above equation is essentially the stochastic equation for a random walk. In general, a random walk is a Markov chain whose next status/location only depends on the current location (the first term in the above equation) and the transition probability (the second term). However, a substantial fraction of the new solutions should be generated by far-field randomization, and their locations should be far enough from the current best solution to make sure that the system will not be trapped in a local optimum [34].

Though the pseudo-code given in many papers is sequential, vectors should be used from the implementation point of view, as vectors are more efficient than loops. A Matlab implementation is given by the author, and it can be downloaded.[2]

The literature on CS is expanding rapidly. Much attention and many recent studies have used CS with a diverse range of applications [7, 11, 27, 37]. Walton et al. improved the algorithm by formulating a modified CS algorithm [27], while Yang and Deb extended it to multiobjective optimization problems [37].

Looking at CS in terms of self-organization, we can see that this swarm-intelligence-based algorithm uses multiple interacting Markov chains by switching between two key branches of global search and local search using Lévy flights as well as random walks so that a balance between global exploration and local exploitation can be achieved during the optimization process.

3.3 Bat Algorithm

A third way of looking at an algorithm, apart from dynamic systems and Markov chains, is by using a varying parameter setting. This idea is used in the bat algorithm; the parameter tuning is essentially achieved by frequency tuning and mimicking the hunting strategy of microbats.

The bat algorithm (BA) is a relatively new metaheuristic, developed by Xin-She Yang in 2010 [40], which was inspired by the echolocation behaviour of microbats. Microbats use a type of sonar, called echolocation, to detect prey, avoid obstacles and locate their roosting crevices in the dark. These bats emit a very loud sound pulse and listen for the echo that bounces back from surrounding objects. Their pulses have varying properties and can be correlated with their hunting strategies, depending on the species. Most bats use short, frequency-modulated signals to sweep through about an octave, while others more often use constant-frequency signals for echolocation. The signal bandwidth varies depending on the species, and is often increased by using more harmonics.

The bat algorithm has three idealized rules:

- All bats use echolocation to sense distance, and they also 'know' the difference between food/prey and background barriers in some magical way.

[2] www.mathworks.com/matlabcentral/fileexchange/29809-cuckoo-search-cs-algorithm.

- Bats fly randomly with velocity $\boldsymbol{v}_i$ at position $\boldsymbol{x}_i$ with a fixed frequency $f_{\min}$, varying wavelength λ and loudness A_0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission $r \in [0, 1]$, depending on the proximity of their target.
- Although the loudness can vary in many ways, we assume that it varies from a large (positive) A_0 to a minimum constant value $A_{\min}$.

Obviously, we have to define the rules of how their positions $\boldsymbol{x}_i$ and velocities $\boldsymbol{v}_i$ in a d-dimensional search space are updated. The new solutions $\boldsymbol{x}_i^t$ and velocities $\boldsymbol{v}_i^t$ at time step t are given by

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta, \tag{10}$$

$$\boldsymbol{v}_i^t = \boldsymbol{v}_i^{t-1} + \left(\boldsymbol{x}_i^{t-1} - \boldsymbol{x}_*\right) f_i, \tag{11}$$

$$\boldsymbol{x}_i^t = \boldsymbol{x}_i^{t-1} + \boldsymbol{v}_i^t, \tag{12}$$

where $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution. Here $\boldsymbol{x}_*$ is the current global best location (solution) which is located after comparing all the solutions among all the n bats at each iteration t. As the product $\lambda_i f_i$ is the velocity increment, we can use f_i (or λ_i) to adjust the velocity change while fixing the other factor λ_i (or f_i), depending on the type of problem of interest. In our implementation, we will use $f_{\min} = 0$ and $f_{\max} = O(1)$, depending on the domain size of the problem of interest. Initially, each bat is randomly assigned a frequency which is drawn uniformly from $[f_{\min}, f_{\max}]$.

For the local search part, once a solution is selected among the current best solutions, a new solution for each bat is generated locally using random walk

$$\boldsymbol{x}_{\text{new}} = \boldsymbol{x}_{\text{old}} + \boldsymbol{\epsilon} A^t, \tag{13}$$

where $\boldsymbol{\epsilon}$ is a random number vector drawn from $[-1, 1]$, while $A^t = \langle A_i^t \rangle$ is the average loudness of all the bats at this time step.

Furthermore, the loudness A_i and the rate r_i of pulse emission have to be updated accordingly as the iterations proceed. As the loudness usually decreases once a bat has found its prey, while the rate of pulse emission increases, the loudness can be chosen as any value of convenience. For simplicity, we can also use $A_0 = 1$ and $A_{\min} = 0$, assuming $A_{\min} = 0$ means that a bat has just found the prey and will temporarily stop emitting any sound. Now we have

$$A_i^{t+1} = \alpha A_i^t, \tag{14}$$

$$r_i^{t+1} = r_i^0 \left[1 - \exp(-\gamma t)\right], \tag{15}$$

where α and γ are constants. In fact, α is similar to the cooling factor of a cooling schedule in the simulated annealing. For any $0 < \alpha < 1$ and $\gamma > 0$, we have

$$A_i^t \to 0, \qquad r_i^t \to r_i^0, \quad \text{as } t \to \infty. \tag{16}$$

In the simplest case, we can use $\alpha = \gamma$, and we have used $\alpha = \gamma = 0.95$ to 0.97 in our simulations.

BA has been extended to the multiobjective bat algorithm (MOBA) by Yang [31], and preliminary results suggest that it is very efficient.

Again looking at BA from the self-organization point of view, the convergence is controlled by loudness and pulse emission rate so that it can explore the vast search space in the earlier stage and then focus on the local exploitation in the more promising regions. Compared with FA and CS, where fixed parameters are used in terms of balancing exploration and exploitation, BA uses a more dynamic approach to balance exploration and exploitation.

4 Engineering Optimization and Applications

Engineering optimization is very diverse with vast collections of case studies, and some case studies require lengthy descriptions to provide sufficient details [6, 10–12, 24]. Here we provide nine case studies as a subset of design optimization benchmarks in engineering and industrial applications.

4.1 Bending Beam Design

Probably the simplest design problem with engineering relevance is the design of a cantilever beam with five different square cross sections with heights/widths of $x_1, x_2, \ldots, x_5$, respectively. The thickness is fixed with $t = 2/3$, and the objective is to minimize [5, 9]

$$f(\boldsymbol{x}) = 0.0624(x_1 + x_2 + x_3 + x_3 + x_4 + x_5), \tag{17}$$

subject to

$$g(\boldsymbol{x}) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \le 0. \tag{18}$$

It is straightforward to use all three algorithms discussed earlier to find the best solution

$$\boldsymbol{x} = (6.0089,\ 5.3049,\ 4.5023,\ 3.5077,\ 2.1504), \tag{19}$$

which gives

$$f_{\min} = 1.33999. \tag{20}$$

4.2 Spring Design

Tensional and/or compressional springs are used widely in engineering. A standard spring design problem has three design variables: the wire diameter w, the mean coil diameter d and the length (or number of coils) L.

The objective is to minimize the weight of the spring, subject to various constraints such as maximum shear stress, minimum deflection and geometrical limits. For a detailed description, please refer to earlier studies [2, 4]. This problem can be written compactly as

$$\text{Minimize} \quad f(\boldsymbol{x}) = (L+2)w^2 d, \tag{21}$$

subject to

$$\begin{aligned} g_1(\boldsymbol{x}) &= 1 - \frac{d^3 L}{71785 w^4} \le 0, \\ g_2(\boldsymbol{x}) &= 1 - \frac{140.45 w}{d^2 L} \le 0, \\ g_3(\boldsymbol{x}) &= \frac{2(w+d)}{3} - 1 \le 0, \\ g_4(\boldsymbol{x}) &= \frac{d(4d-w)}{w^3(12566d - w)} + \frac{1}{5108 w^2} - 1 \le 0, \end{aligned} \tag{22}$$

with the following limits:

$$0.05 \le w \le 2.0, \qquad 0.25 \le d \le 1.3, \qquad 2.0 \le L \le 15.0. \tag{23}$$

Using any of the algorithms discussed earlier, we can easily obtain the same or slightly better solutions than the best solution obtained by [4]:

$$f_* = 0.012665 \quad \text{at } (0.051690, 0.356750, 11.287126), \tag{24}$$

but both CS and FA use significantly fewer evaluations.

4.3 Three-Bar Truss Design

The three-bar truss design is a simple but practical example first presented by Nowcki [22], which requires one to find the optimal cross-sectional areas A_1 and A_2. The problem can be formulated as

$$\text{Minimize} \quad f(A_1, A_2) = (\sqrt{8A_1} + A_2)L, \tag{25}$$

subject to

$$g_1 = \frac{(\sqrt{2}A_1 + A_2)P}{\sqrt{2}A_1^2 + 2A_1A_2} - \sigma \le 0, \tag{26}$$

$$g_2 = \frac{A_2 P}{\sqrt{2}A_1^2 + 2A_1A_2} - \sigma \le 0, \tag{27}$$

$$g_3 = \frac{P}{A_1 + \sqrt{2}A_2} - \sigma \le 0, \tag{28}$$

where $\sigma = 2{,}000$ N/cm^2 is the stress constraint, and $P = 2{,}000$ N/cm^2 is the load. The simple limits are

$$0 \le A_1, A_2 \le 1. \tag{29}$$

Using CS and BA, it is easy to find the optimal solution

$$\boldsymbol{x}_* = (A_1, A_2) = (0.78867, 0.40902), \tag{30}$$

and

$$f_{\min} = 263.97156. \tag{31}$$

4.4 Discrete Beam Design

Reinforced concrete beam designs are relevant in many applications in engineering and construction. One class of beam designs is the discrete beam design, where the dimensions and some design variables can only take discrete values [11, 21]. For example, a very simple design benchmark of a reinforced concrete beam can be written as

$$\text{Minimize} \quad f(A_s, b, h) = 0.6bh + 2.9A_s, \tag{32}$$

subject to

$$g_1 = \frac{h}{b} - 4 \le 0, \tag{33}$$

$$g_2 = \frac{7.375A_s^2}{b} + 180 - A_s h \le 0. \tag{34}$$

However, the area A_s only take values of $\{6.0, 6.16, 6.32, 6.6, 7.0, 7.11, 7.2, 7.8, 7.9, 8.0, 8.4\}$ in^2, and b only takes a value from $\{28, 29, 30, \ldots, 39, 40\}$ and $5 \le h \le 10$ in the continuous domain [21].

By using FA and CS, we have found the best solution

$$f_{\min} = 359.2080, \tag{35}$$

with

$$(A_s, b, h) = (6.32, 34, 8.5), \tag{36}$$

which is better than any solutions found so far in the literature [11].

4.5 Heat Exchanger Design

The heat exchanger design is a problem with six constraints [38], which can be expressed in the simplest case as the following minimization problem with eight design variables:

$$\text{Minimize} \quad f(\boldsymbol{x}) = x_1 + x_2 + x_3, \tag{37}$$

subject to

$$g_1(\boldsymbol{x}) = 0.0025(x_4 + x_6) - 1 \le 0, \tag{38}$$

$$g_2(\boldsymbol{x}) = 0.0025(x_5 + x_7 - x_5) - 1 \le 0, \tag{39}$$

$$g_3(\boldsymbol{x}) = 0.01(x_8 - x_5) - 1 \le 0, \tag{40}$$

$$g_4(\boldsymbol{x}) = 833.33252x_4 + 100x_1 - x_1x_6 - 83333.333 \le 0, \tag{41}$$

$$g_5(\boldsymbol{x}) = 1250x_5 + x_2x_4 - x_2x_7 - 125x_4 \le 0, \tag{42}$$

$$g_6(\boldsymbol{x}) = x_3x_5 - 2500x_5 - x_3x_8 + 1250000 \le 0. \tag{43}$$

For example, using CS with $n = 20$ cuckoos, we can easily find the optimal solution for these eight design variables as

$$x^* = (579.3068, 1359.9708, 5109.9705, 182.0177, \\ 295.6012, 217.9823, 286.4165, 395.6012). \tag{44}$$

4.6 Welded Beam Design

The welded beam design is another standard test problem for constrained design optimization [4, 38]. The problem has four design variables: the width w and length L of the welded area, and the depth d and thickness h of the main beam. The objective is to minimize the overall fabrication cost, under the appropriate constraints of shear stress τ, bending stress σ, buckling load P and maximum end deflection δ.

The problem can be written as

$$\text{minimize} \quad f(\boldsymbol{x}) = 1.10471w^2L + 0.04811dh(14.0 + L), \tag{45}$$

subject to

$$\begin{aligned} g_1(\boldsymbol{x}) &= w - h \le 0, \\ g_2(\boldsymbol{x}) &= \delta(\boldsymbol{x}) - 0.25 \le 0, \\ g_3(\boldsymbol{x}) &= \tau(\boldsymbol{x}) - 13{,}600 \le 0, \\ g_4(\boldsymbol{x}) &= \sigma(\boldsymbol{x}) - 30{,}000 \le 0, \\ g_5(\boldsymbol{x}) &= 0.10471w^2 + 0.04811hd(14+L) - 5.0 \le 0, \\ g_6(\boldsymbol{x}) &= 0.125 - w \le 0, \\ g_7(\boldsymbol{x}) &= 6000 - P(\boldsymbol{x}) \le 0, \end{aligned} \tag{46}$$

where

$$\begin{aligned} &\sigma(\boldsymbol{x}) = \frac{504{,}000}{hd^2}, && Q = 6000\left(14 + \frac{L}{2}\right), \\ &D = \frac{1}{2}\sqrt{L^2 + (w+d)^2}, && J = \sqrt{2}wL\left[\frac{L^2}{6} + \frac{(w+d)^2}{2}\right], \\ &\delta = \frac{65{,}856}{30{,}000hd^3}, && \beta = \frac{QD}{J}, \\ &\alpha = \frac{6000}{\sqrt{2}wL}, && \tau(\boldsymbol{x}) = \sqrt{\alpha^2 + \frac{\alpha\beta L}{D} + \beta^2}, \\ &P = 0.61423 \times 10^6 \frac{dh^3}{6}\left(1 - \frac{d\sqrt{30/48}}{28}\right). \end{aligned} \tag{47}$$

The simple limits or bounds are $0.1 \le L, d \le 10$ and $0.1 \le w, h \le 2.0$. For example, using both CS and FA, we have obtained the following optimal solution:

$$\begin{aligned} \boldsymbol{x}_* &= (w, L, d, h) \\ &= (0.20572963978, 3.47048866563, 9.03662391036, 0.20572963979), \end{aligned} \tag{48}$$

with

$$f(\boldsymbol{x}*)_{\min} = 1.72485230859. \tag{49}$$

This solution is exactly the same as those in the literature [4]

$$f_* = 1.724852 \quad \text{at } (0.205730, 3.470489, 9.036624, 0.205729). \tag{50}$$

We have also solved this problem using BA, and we got exactly the same solution.

4.7 Pressure Vessel Design

Pressure vessels are literally everywhere; some examples are champagne bottles and gas tanks. For a given volume and working pressure, the basic aim of designing a cylindrical vessel is to minimize the total cost. Typically, the design variables are the thickness d_1 of the head, the thickness d_2 of the body, the inner radius r and the length L of the cylindrical section [4, 38]. This is a well-known test problem for optimization and it can be written as

$$\text{minimize} \quad f(\boldsymbol{x}) = 0.6224 d_1 r L + 1.7781 d_2 r^2 + 3.1661 d_1^2 L + 19.84 d_1^2 r, \quad (51)$$

subject to the following constraints:

$$\begin{aligned} g_1(\boldsymbol{x}) &= -d_1 + 0.0193 r \le 0, \\ g_2(\boldsymbol{x}) &= -d_2 + 0.00954 r \le 0, \\ g_3(\boldsymbol{x}) &= -\pi r^2 L - \frac{4\pi}{3} r^3 + 1296000 \le 0, \\ g_4(\boldsymbol{x}) &= L - 240 \le 0. \end{aligned} \quad (52)$$

The simple bounds are

$$0.0625 \le d_1, d_2 \le 99 \times 0.0625, \quad (53)$$

and

$$10.0 \le r, \ L \le 200.0. \quad (54)$$

We have used all three algorithms (FA, CS, BA) to solve this problem, and they all found the same solution $f_* \approx 6,059.714$ at

$$\boldsymbol{x}_* \approx (0.8125, \ 0.4375, \ 42.0984, \ 176.6366), \quad (55)$$

which is the same as the one obtained by Cagnina et al. [4]. This means that the lowest price is about \$6,059.71.

4.8 Gearbox Design

Another important benchmark is the design of a speed reducer which is commonly used in many mechanisms such as a gearbox [14]. This problem involves the optimization of seven variables, including the face width, the number of teeth, the diameter of the shaft and others. All variables are continuous within some limits, except x_3 which only takes integer values. We have

$$\begin{aligned} f(\boldsymbol{x}) = {} & 0.7854 x_1 x_2^2 \left(3.3333 x_3^2 + 14.9334 x_3 - 43.0934\right) \\ & - 1.508 x_1 \left(x_6^2 + x_7^2\right) + 7.4777 \left(x_6^3 + x_7^3\right) + 0.7854 \left(x_4 x_6^2 + x_5 x_7^2\right), \end{aligned} \quad (56)$$

subject to

$$g_1(\boldsymbol{x}) = \frac{27}{x_1 x_2^2 x_3} - 1 \le 0, \tag{57}$$

$$g_2(\boldsymbol{x}) = \frac{397.5}{x_1 x_2^2 x_3^2} - 1 \le 0, \tag{58}$$

$$g_3(\boldsymbol{x}) = \frac{1.93 x_4^3}{x_2 x_3 x_6^4} - 1 \le 0, \tag{59}$$

$$g_4(\boldsymbol{x}) = \frac{1.93 x_5^3}{x_2 x_3 x_7^4} - 1 \le 0, \tag{60}$$

$$g_5(\boldsymbol{x}) = \frac{1.0}{110 x_6^3} \sqrt{\left(\frac{745.0 x_4}{x_2 x_3}\right)^2 + 16.9 \times 10^6} - 1 \le 0, \tag{61}$$

$$g_6(\boldsymbol{x}) = \frac{1.0}{85 x_7^3} \sqrt{\left(\frac{745.0 x_5}{x_2 x_3}\right)^2 + 157.5 \times 10^6} - 1 \le 0, \tag{62}$$

$$g_7(\boldsymbol{x}) = \frac{x_2 x_3}{40} - 1 \le 0, \tag{63}$$

$$g_8(\boldsymbol{x}) = \frac{5 x_2}{x_1} - 1 \le 0, \tag{64}$$

$$g_9(\boldsymbol{x}) = \frac{x_1}{12 x_2} - 1 \le 0, \tag{65}$$

$$g_{10}(\boldsymbol{x}) = \frac{1.5 x_6 + 1.9}{x_4} - 1 \le 0, \tag{66}$$

$$g_{11}(\boldsymbol{x}) = \frac{1.1 x_7 + 1.9}{x_5} - 1 \le 0, \tag{67}$$

where the simple bounds are $2.6 \le x_1 \le 3.6$, $0.7 \le x_2 \le 0.8$, $17 \le x_3 \le 28$, $7.3 \le x_4 \le 8.3$, $7.8 \le x_5 \le 8.4$, $2.9 \le x_6 \le 3.9$, $5.0 \le x_7 \le 5.5$.

The best result in the literature [4] is

$$\boldsymbol{x}_* = (3.5, 0.7, 17, 7.3, 7.8, 3.350214, 5.286683), \tag{68}$$

with $f_{\min} = 2996.348165$.

By using FA and BA as well as CS, we have obtained a new best result:

$$\boldsymbol{x}_* = (3.5, 0.7, 17, 7.3, 7.8, 3.34336449, 5.285351) \tag{69}$$

with the best objective $f_{\min} = 2993.7495888$.

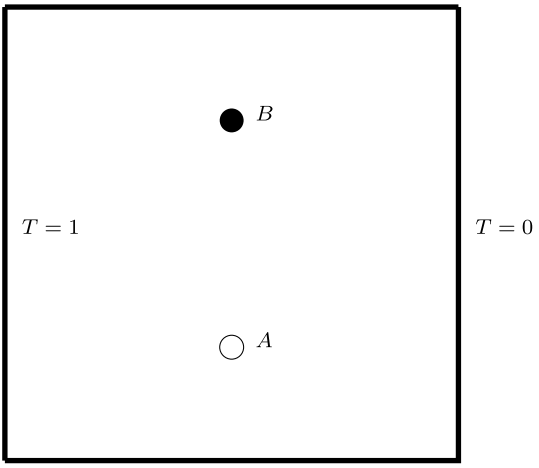


Fig. 1 The topology optimization benchmark to maximize $|T_A - T_B|$

4.9 Simulation-Driven Shape Optimization

Heat management, thus heat transfer modelling, is very important for many electronic applications, especially those using large-scale integrated circuits. In fact, nanoscale heat transfer is a challenging area, and topological optimization for designing nanoscale device is even more challenging [8, 28, 43]. For example, Evgrafov et al. proposed a topology optimization benchmark for a nanoscale heat-conducting system with a size of 150 nm by 150 nm [8]. Heat transfer can occur at many different scales, though smaller scales may be more difficult to control. Now we extend this to a unit of area of 1 mm by 1 mm, and the aim is to distribute two different materials so as to maximize the temperature difference $|T_A - T_B|$ at these two points A and B under the boundary conditions given in Fig. 1 where the top and bottom boundaries are symmetric. Obviously, if there is only one type of material, then $T_A = T_B$ can be expected at the steady state, due to symmetry in the system configuration. However, two types of different materials will change this into a tough shape optimization problem.

Two materials used in the design of the unit area have heat diffusivities of K_1 and K_2, respectively. In addition, $K_1 \gg K_2$. For example, for Si and Mg_2Si, $K_1/K_2 \approx 10$. The domain is continuous, and the objective is to distribute the two materials such that the difference $|T_A - T_B|$ is as large as possible.

By dividing the domain into 40×40 small grids and using CS to search for possible design solutions, an optimal shape and distribution of materials is obtained, as shown in Fig. 2, where Si is shown in light blue and Mg_2Si is shown in red.

For each configuration generated during the search process, the temperature distribution is estimated using the finite difference method by solving the heat conduction equation with varied material conductivities so that the temperature difference at the two fixed points is as large as possible.

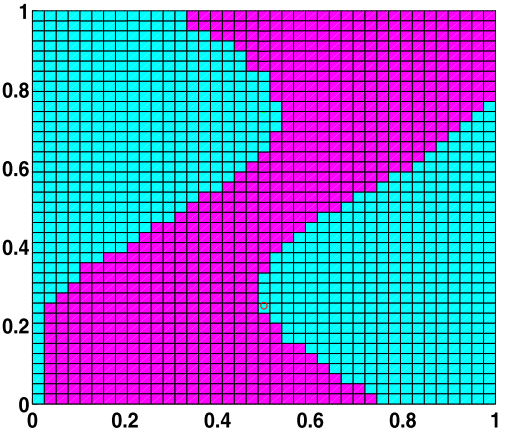


Fig. 2 Optimal topology and distribution of two different materials

5 Challenges and Further Research Topics

Despite the huge success of optimization and extensive applications, many challenging issues must be addressed in the near future. As an optimization process typically involves an optimization algorithm and a simulator, two main issues naturally arise: the efficiency of the algorithm and the efficiency and accuracy of the numerical simulator. Obviously, we try to use the most efficient algorithms available, but the actual efficiency of an algorithm may depend on many factors such as the inner working of the algorithm, the available information (such as objective functions and their derivatives) and implementation details. The associated issue is how to assign the right algorithms to a given problem, which is not easy to solve. In fact, for some highly nonlinear problems, there may not be any efficient algorithm at all. One well-known case is the travelling salesman problem, which is hard in the non-deterministic polynomial-time (NP) sense, that is, NP-hard. There is no efficient algorithm to deal with these types of problems.

The efficiency of a simulator or solver is even more complicated, depending on the actual numerical methods used and the complexity of the problem of interest. Straightforward optimization of a given objective function is not always practical. If the objective function comes from a computer simulation, it may be computationally expensive, noisy or non-differentiable. In such cases, a surrogate-based optimization algorithm may be a very useful alternative [18]. The surrogate model can be typically constructed from the sampled data of the original objective function; however, this surrogate model should reasonably be cheap, smooth and easy to optimize, and yet accurate enough so that it can produce a good prediction of the function's optimum [17]. A challenging issue is how to construct good surrogate models that have good fidelity and yet can save sufficient computational efforts.

On the other hand, it may be no exaggeration to say that metaheuristics have had great success in solving various tough optimization problems. However, there are many important questions which remain unanswered.

First, an important issue to be addressed in any metaheuristic algorithm is how to provide a good balance between local intensification and global diversification [29, 30]. At present, different algorithms uses various techniques and mechanisms with various parameters to control this, which may be far from optimal. Is there any optimal way to achieve this balance? If yes, how? If not, what is the best we can achieve?

Second, it is still only partly understood why different components of heuristics and metaheuristics interact in a coherent and balanced way so that they produce efficient algorithms which converge under the given conditions. For example, why does a balanced combination of randomization and a deterministic component lead to a much more efficient algorithm (than a purely deterministic and/or a purely random algorithm)? How can we measure or test if a balance is reached? How can we prove that the use of memory can significantly increase the search efficiency of an algorithm? Under what conditions?

Finally, most applications in the current literature have been tested against toy problems or small-scale benchmarks with a few design variables or at most problems with several dozen variables. In real-world applications, many design problems in engineering, business and industry may involve thousands or even millions of variables. We have not seen case studies for such large-scale problems in the literature. A crucial issue is that there is no indication that the methodology that works for such toy benchmarks will work equally well for large-scale problems. Apart from the difference in the problem size, there may be other fundamental differences for large-scale problems, and thus the methodology could be very different [33].

Such challenges still remain unresolved, both in theory and in practice. These important issues also provide golden opportunities for researchers to rethink the existing methodology and approaches, perhaps more fundamentally. We can expect that some significant progress will be made in the next ten years.

References

1. Apostolopoulos, T., Vlachos, A.: Application of the firefly algorithm for solving the economic emissions load dispatch problem. Int. J. Comb. **2011**, 523806 (2011). http://www.hindawi.com/journals/ijct/2011/523806.html
2. Arora, J.: Introduction to Optimum Design. McGraw-Hill, New York (1989)
3. Ashby, W.R.: Principles of the self-organizing system. In: Von Foerster, H., Zopf, G.W. Jr. (eds.) Principles of Self-organization: Transactions of the University of Illinois Symposium, pp. 255–278. Pergamon Press, London (1962)
4. Cagnina, L.C., Esquivel, S.C., Coello, C.A.: Solving engineering optimization problems with the simple constrained particle swarm optimizer. Informatica **32**, 319–326 (2008)
5. Chickermane, H., Gea, H.C.: Structural optimization using a new local approximation method. Int. J. Numer. Methods Eng. **39**, 829–846 (1996)
6. Deb, K.: Optimization for Engineering Design. Prentice-Hall, New Delhi (1995)
7. Durgun, I., Yildiz, A.R.: Structural design optimization of vehicle components using cuckoo search algorithm. Mater. Test. **3**, 185–188 (2012)
8. Evgrafov, A., Maute, K., Yang, R.G., Dunn, M.L.: Topology optimization for nano-scale heat transfer. Int. J. Numer. Methods Eng. **77**, 285–300 (2009)

9. Fleury, C., Braibant, V.: Structural optimization: a new dual method using mixed variables. Int. J. Numer. Methods Eng. **23**, 409–428 (1986)
10. Gandomi, A.H., Yang, X.S.: Benchmark problems in structural optimization. In: Koziel, S., Yang, X.S. (eds.) Computational Optimization, Methods and Algorithms. Study in Computational Intelligence, SCI, vol. 356, pp. 259–281. Springer, Berlin (2011)
11. Gandomi, A.H., Yang, X.S., Alavi, A.H.: Cuckoo search algorithm: a meteheuristic approach to solve structural optimization problems. Eng. Comput. doi:10.1007/s00366-011-0241-y (2011). Online first 29 July 2011
12. Gandomi, A.H., Yang, X.S., Talatahari, S., Deb, S.: Coupled eagle strategy and differential evolution for unconstrained and constrained global optimization. Comput. Math. Appl. **63**(1), 191–200 (2012)
13. Gill, P.E., Murray, W., Wright, M.H.: Practical Optimization. Academic Press Inc., London (1981)
14. Golinski, J.: An adaptive optimization system applied to machine synthesis. Mech. Mach. Theory **8**(4), 419–436 (1973)
15. Keller, E.F.: Organisms, machines, and thunderstorms: a history of self-organization, part two. Complexity, emergence, and stable attractors. Hist. Stud. Nat. Sci. **39**, 1–31 (2009)
16. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proc. of IEEE International Conference on Neural Networks, Piscataway, NJ, pp. 1942–1948 (1995)
17. Koziel, S., Yang, X.S.: Computational Optimization and Applications in Engineering and Industry. Springer, Berlin (2011)
18. Koziel, S., Bandler, J.W., Madsen, K.: Quality assessment of coarse models and surrogates for space mapping optimization. Optim. Eng. **9**(4), 375–391 (2008)
19. Koziel, S., Yang, X.S., Zhang, Q.J.: Simulation-Driven Design Optimization and Modeling for Microwave Engineering. Imperial College Press, London (2013)
20. Leifsson, L., Koziel, S.: Multi-fidelity design optimization of transonic airfoils using physics-based surrogate modeling and shape-preserving response prediction. J. Comput. Sci. **1**(2), 98–106 (2010)
21. Liebman, J.S., Khachaturian, N., Chanaratna, V.: Discrete structural optimization. J. Struct. Div. **107**(ST11), 2177–2197 (1981)
22. Nowcki, H.: Optimization in pre-contract ship design. In: Fujita, Y., Lind, K., Williams, T.J. (eds.) Computer Applications in the Automation of Shipyard Operation and Ship Design, vol. 2, pp. 327–338. North-Holland, Elsevier, New York (1974)
23. Prigogine, I., Nicolois, G.: On symmetry-breaking instabilities in dissipative systems. J. Chem. Phys. **46**, 3542–3550 (1967)
24. Ravindran, A., Ragsdell, K.M., Reklaitis, G.V.: Engineering Optimization: Methods and Applications, 2nd edn. Wiley, Hoboken (2006)
25. Sayadi, M.K., Ramezanian, R., Ghaffari-Nasab, N.: A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems. Int. J. Ind. Eng. Comput. **1**, 1–10 (2010)
26. Turing, A.M.: The chemical basis of morphogenesis. Phys. Today **237**, 37–72 (1952)
27. Walton, S., Hassan, O., Morgan, K., Brown, M.R.: Modified cuckoo search: a new gradient free optimization algorithm. Chaos Solitons Fractals **44**(9), 710–718 (2011)
28. Yang, X.S.: Modelling heat transfer of carbon nanotubes. Model. Simul. Mater. Sci. Eng. **13**, 893–902 (2005)
29. Yang, X.S.: Introduction to Computational Mathematics. World Scientific Publishing, Singapore (2008)
30. Yang, X.S.: Engineering Optimization: An Introduction with Metaheuristic Applications. Wiley, New York (2010)
31. Yang, X.S.: Bat algorithm for multi-objective optimisation. Int. J. Bio-Inspir. Comput. **3**(5), 267–274 (2011)
32. Yang, X.S.: Review of meta-heuristics and generalised evolutionary walk algorithm. Int. J. Bio-Inspir. Comput. **3**(2), 77–84 (2011)

33. Yang, X.S.: Nature-inspired metaheuristic algorithms: success and new challenges. J. Comput. Eng. Inf. Technol. **1**, 1–3 (2012). doi:10.4172/2324-9307.1000e101
34. Yang, X.S., Deb, S.: Cuckoo search via Lévy flights. In: Proc. of World Congress on Nature & Biologically Inspired Computing (NaBic 2009), pp. 210–214. IEEE Publications, New York (2009)
35. Yang, X.S., Deb, S.: Engineering optimization by cuckoo search. Int. J. Math. Model. Numer. Optim. **1**(4), 330–343 (2010)
36. Yang, X.S., Deb, S.: Eagle strategy using Lévy walk and firefly algorithms for stochastic optimization. In: Cruz, C., González, R.J., Krasnogor, N., Terrazas, G. (eds.) Nature Inspired Cooperative Strategies for Optimization (NICSO2010). Studies in Computational Intelligence (SCI), vol. 284, pp. 101–111. Springer, New York (2010)
37. Yang, X.S., Deb, S.: Multiobjective cuckoo search for design optimization. Comput. Oper. Res. **40**(6), 1616–1624 (2013). doi:10.1016/j.cor.2011.09.026
38. Yang, X.S., Gandomi, A.H.: Bat algorithm: a novel approach for global engineering optimization. Eng. Comput. **29**(5), 464–483 (2012)
39. Yang, X.S., Koziel, S.: Computational optimization, modelling and simulation—a paradigm shift. Proc. Comput. Sci. **1**(1), 1291–1294 (2010)
40. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: Gonzalez, J.R., et al. (eds.) Nature-Inspired Cooperative Strategies for Optimization (NICSO 2010), vol. 284, pp. 65–74. Springer, Berlin (2010)
41. Yang, X.S., Deb, S., Fong, S.: Accelerated particle swarm optimization and support vector machine for business optimization and applications. In: Networked Digital Technologies 2011. Communications in Computer and Information Science, vol. 136, pp. 53–66 (2011)
42. Yang, X.S., Hossein, S.S., Gandomi, A.H.: Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect. Appl. Soft Comput. **12**(3), 1180–1186 (2012)
43. Zhirnov, V.V., Cavin, R.K., Hutchby, J.A., Bourianoff, G.I.: Limits to binary logic switch scaling—a gedanken model. Proc. IEEE **91**, 1934–1939 (2003)