Published in IET Microwaves, Antennas & Propagation
Received on 18th August 2008
doi: 10.1049/iet-map.2008.0279

ISSN 1751-8725

Distributed fine model evaluation for rapid space-mapping optimisation of microwave structures

S. Koziel[1] *J.W. Bandler*[2,*]

[1]School of Science and Engineering, Reykjavik University, Kringlunni 1, Reykjavik IS-103, Iceland
[2]Simulation Optimization Systems Research Laboratory, Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, Canada L8S 4K1
[*]Bandler Corporation, Dundas, ON, Canada L9H 5E7
E-mail: koziel@ru.is

Abstract: A novel implementation of a space-mapping (SM) algorithm for optimisation of microwave structures and devices is described. The algorithm uses two techniques to speed up the SM optimisation process: the evaluation of the fine model is distributed through independent processing of the fine model response corresponding to consecutive frequency samples using a number of CPUs, and the parameter extraction and surrogate optimisation sub-problems are solved using built-in optimisation capabilities of the coarse model simulator. As a result, the optimisation time of microwave structures can be reduced to values comparable to or smaller than the time necessary for a single fine model evaluation on a single processor. This new implementation can be applied whenever the fine model is evaluated using a frequency-domain simulator. The robustness of this algorithm using microwave design optimisation problems is verified. The efficiency is compared with the standard implementation of the SM algorithm.

1 Introduction

Computer-aided design of electrical networks has been around since the 1960s [1], with optimisation of microwave structures and devices introduced in the late 1960s [2] and the formal electromagnetic optimisation of planar and 3D structures introduced in the early 1990s [3, 4]. The main bottleneck in direct optimisation of microwave structures is the high computational cost of the objective functions that are normally evaluated using full-wave electromagnetic simulators. Fast, circuit-theory-based optimisation using equivalent circuits or analytical or semi-empirical formulas is not accurate enough for practical purposes and the designs obtained with these methods normally need tuning and verification in electromagnetic simulators, although they can serve as reasonable initial approximations of the final design.

Space mapping (SM) addresses the issue of optimisation of expensive functions, also called ‘fine’ models, through iterative optimisation and updating of the surrogate models which are built using cheaper ‘coarse’ models [5–12]. In fact, SM has been originally developed to deal with design problems in microwave engineering. Nowadays, it is gaining popularity in many other engineering areas [13–16].

It is important for the performance of SM algorithms that the coarse model is physically based, that is, that it describes the same physical phenomena as the fine model, typically in a simplified way (e.g. lumped element circuit against full-wave electromagnetic model). This allows us to obtain good alignment between the fine model and the SM surrogate using a small amount of fine model data, as well as an excellent prediction capability of the surrogate. This feature is unavailable for functional surrogate modelling techniques such as polynomial approximation, radial basis functions or Kriging [17–19], which is probably the reason that surrogate-based optimisation methods exploiting functional surrogates [20–23] have not been widely adopted in microwave engineering so far.

Substantial research effort has been devoted to improving the efficiency of SM optimisation. New algorithms and SM surrogate model types have been introduced [6–8, 24]. The convergence properties of SM algorithms have been improved [25, 26]. Assessment methods for automatic or semi-automatic selection of the surrogate model type for a given design problem have been presented [27–29], as well as techniques for creating fast and accurate coarse models [30–32]. These methods aim either at reducing the number of fine model evaluations required to find a satisfactory solution and improving the quality of the final design in terms of satisfying given design specifications, or in reducing the computational overhead of solving the two fundamental sub-problems emerging in the SM algorithm: parameter extraction and surrogate model optimisation.

In this paper, we present a new implementation of an SM optimisation algorithm that uses two techniques to speed-up the SM optimisation process. The first technique is a distributed evaluation of the fine model through independent processing of the fine model response corresponding to consecutive frequency samples using a number of CPUs [33]. It can be applied for fine models evaluated using frequency-domain simulators. The second technique exploits solving the parameter extraction and surrogate optimisation sub-problems using built-in optimisation capabilities of the coarse model simulator [34]. Both techniques have been incorporated in the SMF system [35]. The new implementation of the SM algorithm allows us to reduce the optimisation time of microwave structures to values comparable or smaller than the time necessary for single fine model evaluation on a single processor. The performance of the algorithm is demonstrated using several microwave design optimisation problems. We also provide an efficiency comparison with the standard implementation of the SM algorithm.

2 Formulation of the SM optimisation algorithm

We will use the following notation. Let $\boldsymbol{R}_{\mathrm{f}}(\boldsymbol{x})$ denote the response vector of a fine model corresponding to a design variable vector $\boldsymbol{x}$. In the microwave area, $\boldsymbol{R}_{\mathrm{f}}(\boldsymbol{x})$ may represent the magnitude of a transfer function of a microwave filter at a given set of frequencies. The optimisation problem is formulated as follows

$$\boldsymbol{x}_{\mathrm{f}}^{*} \in \arg\min_{\boldsymbol{x}} U\big(\boldsymbol{R}_{\mathrm{f}}(\boldsymbol{x})\big) \tag{1}$$

where U is a given merit function, for example, a norm or a minimax function [2]; $\boldsymbol{x}_{\mathrm{f}}^{*}$ is the optimal fine model design to be found.

An SM optimisation algorithm generates a sequence of approximate solutions to problem (1), denoted as $\boldsymbol{x}^{(i)}$, $i = 0, 1, 2, \ldots$, and a family of surrogate models $\boldsymbol{R}_{\mathrm{s}}^{(i)}$, so that we have

$$\boldsymbol{x}^{(i+1)} \in \arg\min_{\boldsymbol{x}} U\Big(\boldsymbol{R}_{\mathrm{s}}^{(i)}(\boldsymbol{x})\Big) \tag{2}$$

Let $\boldsymbol{R}_{\mathrm{c}}$ denote the response vector of the coarse model that describes the same object as the fine model: less accurate but much faster to evaluate. Surrogate models are constructed from the coarse model so that the misalignment between $\boldsymbol{R}_{\mathrm{s}}^{(i)}$ and the fine model is minimised. $\boldsymbol{R}_{\mathrm{s}}^{(i)}$ is defined as [7]

$$\boldsymbol{R}_{\mathrm{s}}^{(i)}(\boldsymbol{x}) = \bar{\boldsymbol{R}}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{p}^{(i)}) \tag{3}$$

where $\bar{\boldsymbol{R}}_{\mathrm{s}}$ is a generic SM surrogate model, that is, the coarse model composed with suitable transformations, whereas

$$\boldsymbol{p}^{(i)} \in \arg\min_{\boldsymbol{p}} \sum_{k=0}^{i} w_{i.k} \| \boldsymbol{R}_{\mathrm{f}}(\boldsymbol{x}^{(k)}) - \bar{\boldsymbol{R}}_{\mathrm{s}}(\boldsymbol{x}^{(k)}, \boldsymbol{p}) \| \tag{4}$$

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.

There is a number of SM surrogate models available [6–8], which can be roughly divided into four groups:

i. Models based on a (usually linear) distortion of the coarse model parameter space, for example, input SM of the form $\bar{\boldsymbol{R}}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{p}) = \bar{\boldsymbol{R}}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{B}, \boldsymbol{c}) = \boldsymbol{R}_{\mathrm{c}}(\boldsymbol{B} \cdot \boldsymbol{x} + \boldsymbol{c})$ [5],

ii. Models based on a distortion of the coarse model response, for example, output SM of the form $\bar{\boldsymbol{R}}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{p}) = \bar{\boldsymbol{R}}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{d}) = \boldsymbol{R}_{\mathrm{c}}(\boldsymbol{x}) + \boldsymbol{d}$ or of the form $\bar{\boldsymbol{R}}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{p}) = \bar{\boldsymbol{R}}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{A}) = \boldsymbol{A} \cdot \boldsymbol{R}_{\mathrm{c}}(\boldsymbol{x})$ [7],

iii. Implicit SM, where the parameters used to align the surrogate with the fine model are separate from the design variables, that is, $\bar{\boldsymbol{R}}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{p}) = \bar{\boldsymbol{R}}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}_{\mathrm{p}}) = \boldsymbol{R}_{\mathrm{c}.i}(\boldsymbol{x}, \boldsymbol{x}_{\mathrm{p}})$, with $\boldsymbol{R}_{\mathrm{c}.i}$ being the coarse model dependent on both design variables $\boldsymbol{x}$ and so-called pre-assigned parameters $\boldsymbol{x}_{\mathrm{p}}$ (e.g. dielectric constant, substrate height) that are normally fixed in the fine model but can be freely changed in the coarse model [36] and

iv. Custom models exploiting parameters characteristic to a given design problem; the most characteristic example is the so-called frequency SM $\bar{\boldsymbol{R}}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{p}) = \bar{\boldsymbol{R}}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{F}) = \boldsymbol{R}_{\mathrm{c.f}}(\boldsymbol{x}, \boldsymbol{F})$ [6], where $\boldsymbol{R}_{\mathrm{c.f}}$ is a frequency-mapped coarse model, that is, the coarse model evaluated at frequencies different from the original frequency sweep for the fine model, according to the mapping $\omega \rightarrow f_1 + f_2\omega$, with $\boldsymbol{F} = [f_1 f_2]^{\mathrm{T}}$.

The SM optimisation algorithm flow can be described as follows:

Step 1: Set $i = 0$; choose the initial solution $\boldsymbol{x}^{(0)}$;

Step 2: Evaluate the fine model to find $\boldsymbol{R}_{\mathrm{f}}(\boldsymbol{x}^{(i)})$;

Step 3: Obtain the surrogate model $\boldsymbol{R}_s^{(i)}$ using (3) and (4);

Step 4: Given $\boldsymbol{x}^{(i)}$ and $\boldsymbol{R}_s^{(i)}$, obtain $\boldsymbol{x}^{(i+1)}$ using (2);

Step 5: If the termination condition is not satisfied go to Step 2; else terminate the algorithm.

Typically, $\boldsymbol{x}^{(0)} = \arg\min\{\boldsymbol{x}{:}U(\boldsymbol{R}_c(\boldsymbol{x}))\}$, that is, it is the optimal solution of the coarse model, which is the best initial design we normally have at our disposal.

Usually, the algorithm is terminated when it converges (i.e. $\|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(i-1)}\|$ and/or $\|\boldsymbol{R}_f(\boldsymbol{x}^{(i)}) - \boldsymbol{R}_f(\boldsymbol{x}^{(i-1)})\|$ are smaller than user-defined values) or when the maximum number of iterations (or fine model evaluations) is exceeded. If the surrogate model is a sufficiently good representation of the fine model [27], the SM algorithm typically requires a few fine model evaluations to yield a satisfactory solution, however, we cannot expect the final solution to be a local optimum of the fine model in general, unless, for example, first-order consistency conditions between the surrogate and the fine model are ensured (which requires exploiting fine model sensitivity data [7]) and convergence safeguards such as trust region methods are used [7].

As mentioned in the introduction, there has been a continuous effort to make SM more and more efficient in two aspects, that is, reducing the computational complexity of the optimisation process and improving the quality of the final design produced by the algorithm. It follows from the flowchart presented above that the reduction of the computational cost of the SM algorithm can be obtained through:

i. Reduction of the number of fine model evaluations,

ii. Reduction of the computational overhead of parameter extraction and surrogate model optimisation or

iii. Decreasing the evaluation time for the fine model.

The first two goals can be obtained by improving the convergence properties of the SM algorithm, using more accurate and computationally cheaper coarse models, as well as proper selection of the SM type for a given design problem. All these possibilities have been explored recently as described in the introduction.

The last goal, described in the next section, can be realised by a distributed evaluation of the fine model.

3 Distributed evaluation of the fine model in the SMF system

A distributed evaluation of the fine model has been implemented within the SMF system, a user-friendly SM software engine, allowing automated SM optimisation of microwave devices and circuits [35].

3.1 Architecture of the distributed model evaluation

Distributed evaluation of the fine model is realised through independent processing of the fine model responses corresponding to consecutive frequency samples using a number of machines. Thus, it can be applied for models using frequency-domain simulators. Because parallelisation is implemented internally in the SMF system, it works regardless of whether the fine model simulator has a multi-processor analysis capability or not.

Fig. 1 shows the flowchart of the distributed fine model evaluation [33]. Evaluation is performed by the main SMF copy and by n distributed evaluation clients (SMFDs) running on separate processors. Suppose that the fine model is evaluated at m frequency points, $f_1, f_2, \ldots, f_m$. This set of frequencies is divided into K sub-bands, B_1 to B_K. In particular, the sub-bands may consist of single frequency samples.

The information about the design variable vector $\boldsymbol{x}$ and frequency sub-bands is put into a so-called order set. Orders are picked up and processed by both the main SMF copy and by the SMFD clients and the results are exported into the results set, which is periodically checked by the main SMF program. Once all orders are processed and the corresponding responses are in the response set, the complete fine model response is returned.

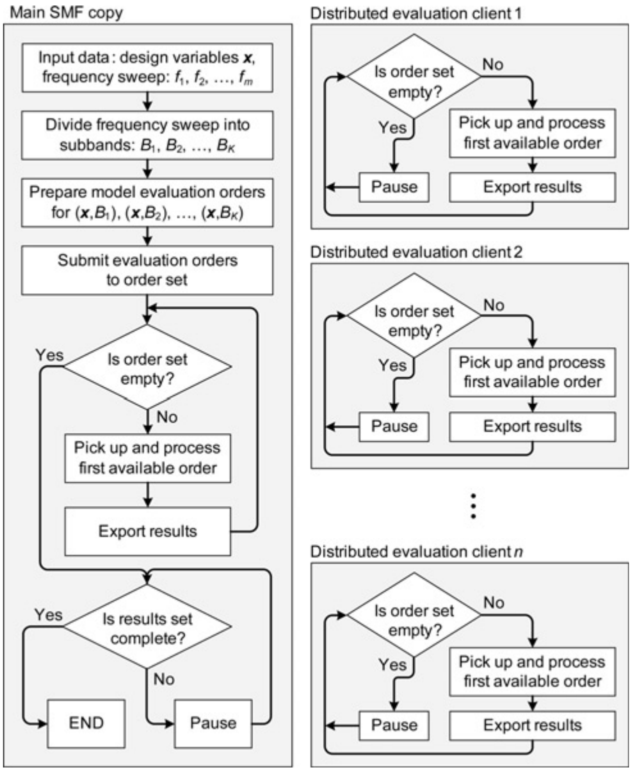
Main SMF copy
Input data: design variables x, frequency sweep: f1, f2, ..., fm
Divide frequency sweep into subbands: B1, B2, ..., BK
Prepare model evaluation orders for (x,B1), (x,B2), ..., (x,BK)
Submit evaluation orders to order set
Yes
Is order set empty?
No
Pick up and process first available order
Export results
Yes
Is results set complete?
No
END
Pause
Distributed evaluation client 1
Is order set empty?
No
Yes
Pick up and process first available order
Pause
Export results
Distributed evaluation client 2
Is order set empty?
No
Yes
Pick up and process first available order
Pause
Export results
Distributed evaluation client n
Is order set empty?
No
Yes
Pick up and process first available order
Pause
Export results

Figure 1 *Flowchart of distributed model evaluation in the SMF system [33]*

Fig. 2 shows the architecture of the distributed model evaluation. All the information about the model, including the data allowing SMF and the SMFDs to prepare simulator input files, call the simulator and format the output data as well as the evaluation vector $\boldsymbol{x}$ and frequency sub-band, is stored in the so-called order files. If SMF requests model evaluation, a number of order files corresponding to the number of frequency sub-bands as described before are generated and copied to a separate folder accessible by all SMFD clients. SMF and the SMFD clients pick up available order files and, after processing them, return the results to a results folder. Each SMFD client uses a separate working folder for temporary files. All the folders may reside in a designated directory on a local network drive or in a file system of a computing cluster.

Communication between SMF, the SMFDs and the folders is realised through the SSH protocol. It should be noted that the described distributed evaluation implementation has cross-platform capability and it may involve any number of machines running under Windows or Linux, both workstations and PCs, at the same time.

3.2 Parallel efficiency of distributed model evaluation process

In the ideal world, assuming that the main SMF program and n SMFD clients are used in the distributed model evaluation process and all of them are running on identical processors, the computation time should be $n+1$ times smaller than the evaluation time on a single processor. In practice, this is never the case because of the following factors:

i. In order to obtain maximum possible efficiency the number K of frequency sub-bands should be an integer multiplier of the number of processors $n+1$, which may not be the case;

ii. The CPU type and speed, and, consequently, evaluation time of the order files, may be different for different processors;

iii. There is some overhead related to communication between SMF and the SMFDs and the designated folders;

iv. There may be additional overhead related to the fact that some actions which would normally be done once, for example, meshing of the structure, might be performed for each frequency sub-band separately by each SMFD client.

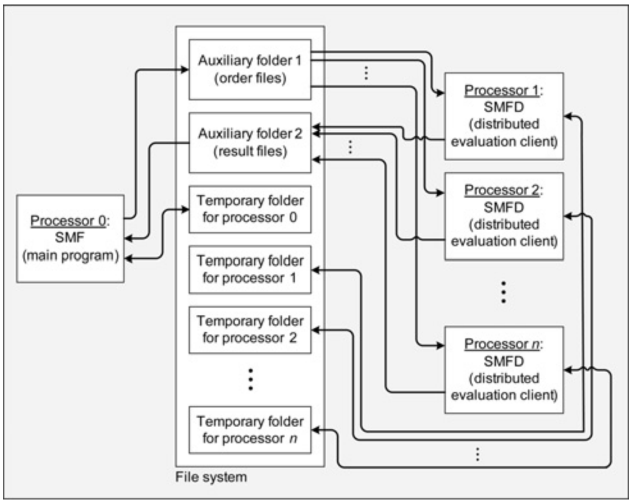

Figure 2 *Architecture of distributed model evaluation in the SMF system [33]*

The first factor plays the crucial role and the speed-up s that can be obtained with our method, neglecting factors (ii), (iii) and (iv), is given by

$$s = \frac{K}{\lceil K/(n+1) \rceil} \tag{5}$$

where $\lceil . \rceil$ denotes the ceiling function.

The parallel efficiency ε is defined as the speed-up divided by the number of processors [37], that is

$$\varepsilon = \frac{s}{n+1} \tag{6}$$

For example, if we have 30 sub-bands and 8 processors, the speed-up is 7.5 and the parallel efficiency is about 94%.

Because of factors (ii), (iii) and (iv), the actual parallel efficiency is smaller, typically between 60 and 90%. We assume here that the number of processors is properly related to the number of frequency samples, that is, the speed-up s (5) is sufficiently high, for example, 90% and more.

4 Improving efficiency of solving parameter extraction and surrogate model optimisation sub-problems

The method described in Section 3 allows us to reduce the evaluation time of the fine model, which is the main factor limiting the speed of SM optimisation. However, if this evaluation time is being reduced, and the coarse model is implemented in a circuit simulator that is being called by an external optimisation routine while solving parameter extraction (4) and surrogate model optimisation (2), which is the case in standard implementations of SM algorithms in the microwave area, the time necessary to solve (2) and (4), normally negligible, becomes a substantial component of the total SM optimisation time. Here, we assume that a coarse model is implemented in Agilent ADS [38]. ADS can be considered as the primary coarse model evaluator in the microwave area because it is widely used and it allows convenient and straightforward creation of coarse models for many microwave structures and devices.

In this section, we describe the problems related to using ADS as a 'black box' coarse model evaluator in an SM algorithm and how the built-in optimisation capabilities of ADS can be exploited to reduce the computational time of solving the parameter extraction and surrogate optimisation problems.

4.1 Standard implementation

The standard implementation of the SM optimisation algorithm assumes that both the parameter extraction and surrogate model optimisation sub-problems are solved using appropriate optimisation routines that make calls to the coarse model simulator each time the coarse model has to be evaluated. In this work, we use routines provided with Matlab's Optimisation Toolbox [39], in particular fmincon or lsqnonlin in case of parameter extraction, and fminimax for surrogate optimisation. Each time we invoke an ADS simulation, CPU clock cycles are consumed on allocating memory, loading the simulator, verifying license, loading the input file, parsing the input file, simulating the circuit, exporting the response etc. Although the circuit simulation is usually fast for a single design, calling ADS simulation repeatedly will generate a significant overhead that cannot be neglected in the SM optimisation process.

Fig. 3 shows the interaction between the optimisation routine and ADS within the standard implementation of the SM algorithm. Assuming that the optimisation process (either parameter extraction or surrogate model optimisation) requires k evaluations of the coarse model, the total optimisation time would be $k(t_o + t_s)$, where t_o is the overhead time, and t_s is the circuit simulation time for a single design. In our case, t_s is typically a few milliseconds, whereas t_o might be as large as a few seconds, so that we have $t_s < < t_o$.

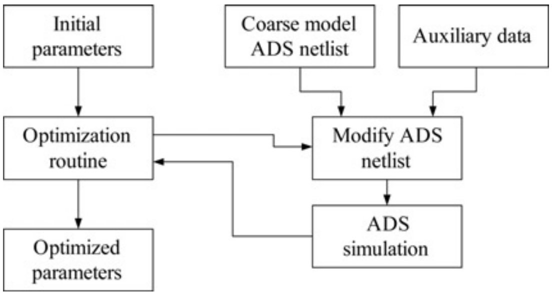

Figure 3 *Interaction between the optimisation routine and ADS for the standard implementation of the SM algorithm*

Initial/optimised parameters are either SM parameters (in the case of parameter extraction) or design variables (in the case of surrogate model optimisation)
Auxiliary data include design specifications, SM formulas and setup information
ADS is repeatedly called as an external function evaluator with the coarse model netlist being modified to correspond to the design currently considered by the optimisation routine

Having in mind that typical surrogate optimisation requires dozens or hundreds, while parameter extraction (especially multi-point) sometimes even thousands of coarse model evaluations, the overhead time may be quite significant.

4.2 Inside-ADS parameter extraction and surrogate model optimisation

To reduce the parameter extraction and surrogate optimisation time we take advantage of the ADS multipoint simulation and built-in optimisation capabilities [34]. (ADS implements various optimisation routines, including Quasi–Newton and minimax, and allows simulation at multiple designs specified by DAC components.) In particular, it is possible to solve the whole parameter extraction and surrogate optimisation sub-problems inside the ADS simulator. Compared with the traditional way, the optimisation loop is moved into ADS. Since the loop is inside ADS, the SM algorithm only prepares the modified ADS netlist and initiates one call to ADS for the entire optimisation process. As the optimisation takes a lot of surrogate model simulations, a large amount of time is saved.

The netlist originally containing only the coarse model implementation in the traditional approach, is enhanced by DAC components importing multiple designs and corresponding fine model responses, by VAR components incorporating SM equations and matrices, by optimisation GOAL components specifying matching goals between fine and surrogate models, and, finally, by optimisation engine OPTIM that searches for the optimal solution for parameter extraction.

Note that netlist modifications required for parameter extraction and surrogate model optimisation are different as in the first case we solve a least-square optimisation problem (4) with respect to the SM parameters and with the matching error computed at multiple designs at the same time, whereas in the latter case, we typically solve a (minimax) problem (2) for a space-mapped coarse model with respect to the design variables.

Fig. 4 shows the flowchart for executing parameter extraction (surrogate optimisation) according to the inside-ADS optimisation approach. The coarse model netlist is modified only once, and only a single ADS call is required. Again, assuming that the optimisation process (either parameter extraction or surrogate model optimisation) requires k evaluations of the coarse model, the total optimisation time would be $t_o' + k \cdot t_s'$, where t_o' is the overhead time, and t_s' is the circuit simulation time for a modified design. While t_o' and t_s' are, in general, different from t_o and t_s, they are of the same order of magnitude (milliseconds and seconds, respectively), so that the time savings by exploiting the inside-ADS optimisation are substantial as shown in Section 5.

doi: 10.1049/iet-map.2008.0279

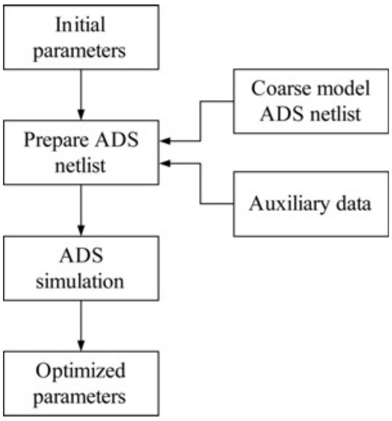

Figure 4 *Inside-ADS parameter extraction and surrogate optimisation flowchart*

Initial/optimised parameters are either SM parameters (in the case of parameter extraction) or design variables (in the case of surrogate model optimisation)
Auxiliary data include design specifications, SM formulas and setup information
ADS is called only once with the coarse model netlist modified to contain all the data necessary to solve a given sub-problem

5 Examples

In this section, we consider two examples of microwave design problems that are solved using three methods: (i) a standard implementation of an SM algorithm, (ii) an SM algorithm with distributed fine model evaluation (Section 3) and (iii) the SM algorithm with distributed fine model evaluation and inside-ADS parameter extraction and surrogate model optimisation (Section 4).

5.1 Bandpass microstrip filter with open stub inverter

Consider the bandpass microstrip filter with open stub inverter [40] shown in Fig. 5. The design parameters are $x = [L_1\ L_2\ L_3\ S_1\ S_2\ W_1]^T$. The fine model is simulated in FEKO [41]. The design specifications are $|S_{21}| \leq -20$ dB for 1.5 GHz $\leq \omega \leq$ 1.8 GHz, $|S_{21}| \geq -3$ dB for 1.95 GHz $\leq \omega \leq$ 2.05 GHz and $|S_{21}| \leq -20$ dB for 2.2 GHz $\leq \omega \leq$ 2.5 GHz, where S_{21} is the complex transmission coefficient between the input and output ports. The model response is the evaluation of $|S_{21}|$ at 41 frequency points uniformly distributed in the interval 1.5 to 2.5 GHz. This number of frequency samples was selected to capture the filter response with sufficient accuracy. The coarse model is the circuit model implemented in Agilent ADS [38] shown in Fig. 6. The initial design is the coarse model optimal solution $x^{(0)} = [25.00\ \ 5.00\ \ 1.221\ \ 0.652\ \ 0.187\ \ 0.100]^T$ mm (fine model specification error +15.7 dB).

We use the surrogate model of the form $\bar{R}_s(x, p) = \bar{R}_s(x, F, c, d) = R_{c.f}(x + c) + d$, which corresponds to input, frequency [6], and output SM, where $R_{c.f}$ is the coarse model evaluated at frequencies different from the original frequency sweep for the fine model, according to the mapping $\omega \rightarrow f_1 + f_2 \cdot \omega$, with $F = [f_1 f_2]^T$; the output parameter d is calculated as $d = R_f(x) - R_{c.f}(x + c)$ after the extractable parameters c and F are known. Fig. 7 shows the fine model response at $x^{(0)}$ (dashed line) as well as the response of the optimised fine model (solid line) obtained after four SM iterations ($x^{(4)} = [23.64\ 5.00\ 1.00\ 0.742\ 0.189\ 0.100]^T$ mm; specification error is −1.9 dB).

Table 1 shows a comparison of the optimisation time for the three implementations of the SM algorithm. For the standard implementation, most of the computational cost comes from the fine model evaluation (about 36 min per evaluation on a Pentium D 3.4 GHz processor). The contribution of coarse model evaluation time because of solving the parameter extraction and surrogate optimisation is relatively small.

Our SM algorithm with distributed fine model evaluation uses 14 processors (1 Pentium D 3.4 GHz for SMF, and 13

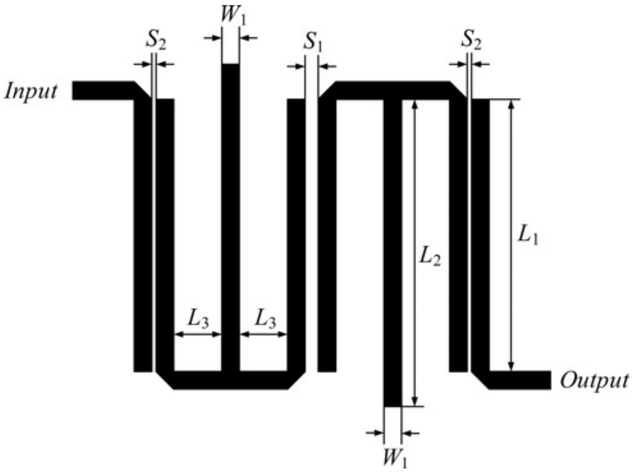

Figure 5 *Geometry of the bandpass filter with open stub inverter [40]*

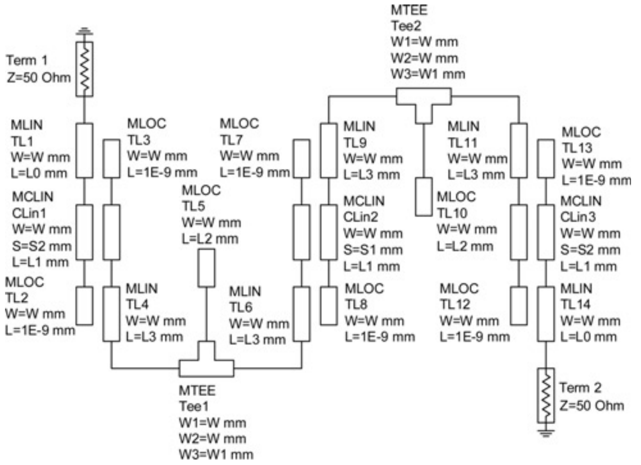

Figure 6 *Coarse model of the bandpass filter with open stub inverter (Agilent ADS)*

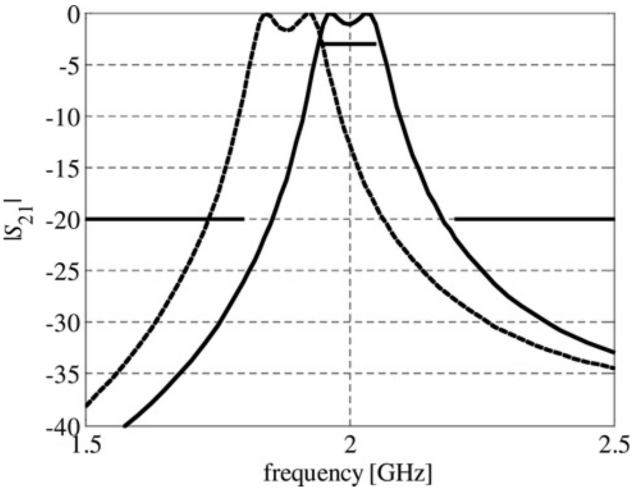

Figure 7 *Initial (dashed line) and optimised (solid line) $|S_{21}|$ against frequency for the bandpass filter with open stub inverter*

nodes of the computing cluster containing Dual Core AMD 2 GHz processors and Intel Xeon 3.06 GHz processors for the SMFDs), which gives a very good speed-up (5) of 13.7 and a parallel efficiency of more than 97% (calculated assuming that the processors' speed is the same). The actual distributed evaluation time is about 3 min 10 s, which gives a parallel efficiency of about 81%. Out of the factors indicated in Section 3.2, the principal reason for the above evaluation time is, in our case, the difference in CPU speed between the processors (the actual number of processors of each of the two types indicated above involved in the evaluation process is unknown because it is automatically determined by the task scheduler of the computing cluster).

When the distributed evaluation of the fine model is exploited, the total fine model evaluation time is reduced from 181 min to only 16 min. As a result, the computational cost of solving the parameter extraction and surrogate optimisation sub-problems is now almost 80% of the total optimisation cost. The application of inside-ADS parameter extraction and surrogate optimisation allows further reduction of the SM optimisation time to only 19 min, which is slightly more than half of the time necessary to evaluate the fine model on a single processor.

To emphasise the advantages of SM over direct optimisation, we also optimised the filter using Matlab's fminimax routine [39], a gradient-based algorithm. This direct optimisation fails to find the fine model optimum. The best solution found in 150 fine model evaluations (some four days of CPU time on a single processor) corresponds to a specification error of about +4 dB.

It should also be noted that the difference in solving the parameter extraction and surrogate optimisation between the standard implementation and inside-ADS optimisation is not as large as might be expected by comparing the simulation time t_s and the overhead time t_o (milliseconds against a few seconds). This is because the ADS and Matlab optimisation routines are different, and it is not possible to access certain control parameters for the ADS optimisation routines (e.g. only the maximum number of iterations can be specified as a termination condition). The number of objective function evaluations necessary to complete the process is generally larger for ADS than for Matlab.

5.2 Third-order Chebyshev bandpass filter

Consider the third-order Chebyshev bandpass filter [42] shown in Fig. 8. The design parameters are $\boldsymbol{x} = [L_1\ L_2\ S_1\ S_2\ W_1\ W_2]^T$ mm. The fine model $\boldsymbol{R}_f$ is simulated in Sonnet ***em*** [43] with a fine grid of 0.1 mm × 0.02 mm. The design specifications are $|S_{21}| \geq -3$ dB for $1.8\text{ GHz} \leq \omega \leq 2.2\text{ GHz}$, and $|S_{21}| \leq -20$ dB for $1.0\text{ GHz} \leq \omega \leq 1.6\text{ GHz}$ and $2.4\text{ GHz} \leq \omega \leq 3.0\text{ GHz}$. The model response is the evaluation of $|S_{21}|$ at 41 frequency points uniformly distributed in the interval 1.0 to 3.0 GHz. The coarse model is the circuit model implemented in Agilent ADS [38] shown in Fig. 9. The initial design is the coarse model optimal solution $\boldsymbol{x}^{(0)} = [14.7\ \ 15.3\ \ 0.62\ \ 0.50\ \ 0.20\ \ 0.20]^T$ mm (fine model specification error +7.2 dB).

We use the surrogate model of the form $\bar{\boldsymbol{R}}_s(\boldsymbol{x}, \boldsymbol{p}) = \bar{\boldsymbol{R}}_s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{d}) = \boldsymbol{R}_c(\boldsymbol{x} + \boldsymbol{c}) + \boldsymbol{d}$, which corresponds to input and output SM; the output parameter $\boldsymbol{d}$ is calculated as $\boldsymbol{d} = \boldsymbol{R}_f(\boldsymbol{x}) - \boldsymbol{R}_c(\boldsymbol{x} + \boldsymbol{c})$ after the extractable parameter $\boldsymbol{c}$ is known. Fig. 10 shows the fine model response at $\boldsymbol{x}^{(0)}$ (dashed line) as well as the response of the optimised fine model (solid line) obtained after five iterations of the SM

Table 1 Bandpass filter with open stub inverter: optimisation time for the three implementations of SM

SM algorithm	Total optimisation time, min	Fine model evaluation time	Parameter extraction and surrogate model optimisation time	Time savings
standard implementation	241	181 min (75%)	60 min (25%)	–
distributed fine model evaluation	76	16 min (21%)	60 min (79%)	68%
distributed fine model evaluation and inside-ADS optimisation	19	16 min (84%)	3 min (16%)	92%

doi: 10.1049/iet-map.2008.0279

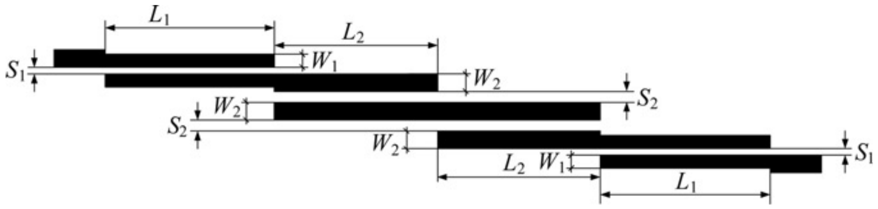

Figure 8 *Third-order Chebyshev bandpass filter: Physical structure [43]*

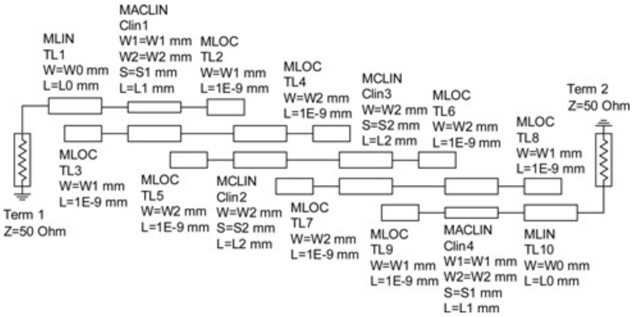

Figure 9 *Third-order Chebyshev bandpass filter: Coarse model (Agilent ADS)*

algorithm with distributed fine model evaluation and inside-ADS optimisation ($\boldsymbol{x}^{(5)} = [15.1 \quad 14.5 \quad 0.54 \quad 0.74 \quad 0.20 \quad 0.60]^T$ mm; specification error is −1.9 dB). It should be noted that the SM algorithm using the standard implementation of the parameter extraction and surrogate model optimisation required four iterations to arrive at the same solution, which is merely because of the fact that Matlab and ADS are using different optimisation routines.

Table 2 shows a comparison of the optimisation time for the three implementations of the SM algorithm. For the standard implementation, most of the computational cost comes from the fine model evaluation (about 27 min per evaluation on a Pentium D 3.4 GHz processor). The contribution of the coarse model evaluation time because of solving the parameter extraction and surrogate optimisation is relatively small.

We use the same computing facility (14 processors) to execute the SM algorithm with distributed fine model evaluation. The speed-up (5) is 13.7 and parallel efficiency is 97% (calculated as if all the processors were the same).

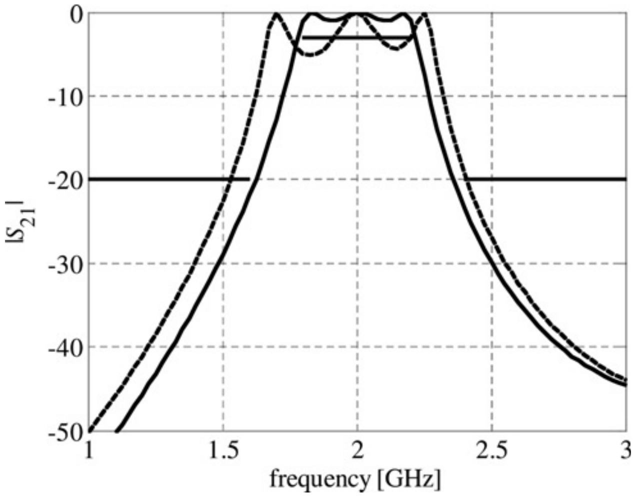

Figure 10 *Initial (dashed line) and optimised (solid line) $|S_{21}|$ against frequency for the bandpass filter with open stub inverter*

The actual distributed evaluation time is about 2 min 50 s, which gives a parallel efficiency of about 68%. As before, the main reason for that is the difference in CPU speed between the processors.

Distributed evaluation of the fine model allows us to reduce the total fine model evaluation time from 135 min to just 14 min, which results in a 74% reduction of the SM optimisation time. The relative cost of solving the parameter extraction and surrogate optimisation sub-problems increases, however, to almost 70% of the total optimisation cost. When the inside-ADS parameter extraction and surrogate optimisation is also exploited, the total optimisation time can be further reduced to only 20 min (time savings of 88% with respect to a standard

Table 2 Third-order Chebyshev bandpass filter: optimisation time for the three implementations of SM

SM algorithm	Total optimisation time, min	Fine model evaluation time	Parameter extraction and surrogate model optimisation time	Time savings
standard implementation	163	135 min (83%)	28 min (17%)	–
distributed fine model evaluation	42	14 min (33%)	28 min (67%)	74%
distributed fine model evaluation and inside-ADS optimisation	20	17 min (85%)	3 min (15%)	88%

implementation) even though one extra fine model evaluation is necessary.

As in the previous example, we attempted direct optimisation of the filter. Because the fine model is available on a discrete grid, gradient-based methods cannot be used. Instead, we used a grid search procedure which yielded a solution comparable with that found by SM (specification error -1.8 dB), however, the number of fine model evaluations required is 168 (over three days of CPU time on a single processor).

6 Conclusion

A computationally efficient implementation of our SM optimisation algorithm is presented. It exploits the distributed evaluation of the fine model and solving the parameter extraction and surrogate optimisation sub-problems inside the coarse model simulator. Both techniques applied together allow substantial reduction of the SM optimisation time in comparison with the standard implementation. The optimisation process is automated and carried out in our SMF system. Through microwave design problems we demonstrate that it is possible to complete the optimisation in less time than necessary to evaluate the fine model on a single processor.

7 Acknowledgments

The authors thank Sonnet Software, Inc., Syracuse, NY, for ***em***$^{\text{TM}}$ and Agilent Technologies, Santa Rosa, CA, for making ADS available. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grants RGPIN7239-06 and STPGP336760-06, and by Bandler Corporation.

8 References

[1] TEMES G.C., CALAHAN D.A.: 'Computer-aided network optimization the state-of-the-art', *Proc. IEEE*, 1967, **55**, (11), pp. 1832–1863

[2] BANDLER J.W.: 'Optimization methods for computer-aided design', *IEEE Trans. Microw. Theory Tech.*, 1969, **MTT-17**, (8), pp. 533–552. Also in *Microwave Integrated Circuits*, FREY J. (ED.) (Artech House, 1975)

[3] BANDLER J.W., BIERNACKI R.M., CHEN S.H., SWANSON D.G. JR., YE S.: 'Microstrip filter design using direct EM field simulation', *IEEE Trans. Microw. Theory Tech.*, 1994, **42**, (7), pp. 1353–1359

[4] BANDLER J.W., BIERNACKI R.M., CHEN S.H., *ET AL.*: 'Electromagnetic design of high-temperature superconducting microwave filters', *Int. J. Microw. Millim.-Wave Comput.-Aided Eng.*, 1995, **5**, (5), pp. 331–343

[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.*, 1994, **4**, (12), pp. 536–544

[6] BANDLER J.W., CHENG Q.S., DAKROURY S.A., *ET AL.*: 'Space mapping: the state of the art', *IEEE Trans. Microw. Theory Tech.*, 2004, **52**, (1), pp. 337–361

[7] KOZIEL S., BANDLER J.W., MADSEN K.: 'A space mapping framework for engineering optimization: theory and implementation', *IEEE Trans. Microw. Theory Tech.*, 2006, **54**, (10), pp. 3721–3730

[8] ECHEVERRIA D., HEMKER P.W.: 'Space mapping and defect correction', *CMAM: Int. Math. J. Comput. Methods Appl. Math.*, 2005, **5**, (2), pp. 107–136

[9] ISMAIL M.A., SMITH D., PANARIELLO A., WANG Y., YU M.: 'EM-based design of large-scale dielectric-resonator filters and multiplexers by space mapping', *IEEE Trans. Microw. Theory Tech.*, 2004, **52**, (1), pp. 386–392

[10] AMARI S., LEDREW C., MENZEL W.: 'Space-mapping optimization of planar coupled-resonator microwave filters', *IEEE Trans. Microw. Theory Tech.*, 2006, **54**, (5), pp. 2153–2159

[11] 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.*, 2006, **54**, (12), pp. 4528–4537

[12] ZHANG X.J., FANG D.G.: 'Using circuit model from layout-level synthesis as coarse model in space mapping and its application in modelling low-temperature ceramic cofired radio frequency circuits', *Microw. Antennas Propag., IET*, 2007, **1**, (4), pp. 881–886

[13] CREVECOEUR G., DUPRE L., VAN DE WALLE R.: 'Space mapping optimization of the magnetic circuit of electrical machines including local material degradation', *IEEE Trans. Magn.*, 2007, **43**, (6), pp. 2609–2611

[14] LEARY S.J., BHASKAR A., KEANE A.J.: 'A constraint mapping approach to the structural optimization of an expensive model using surrogates', *Optim. Eng.*, 2001, **2**, (4), pp. 385–398

[15] REDHE M., NILSSON L.: 'Using space mapping and surrogate models to optimize vehicle crashworthiness design'. 9th AIAA/ISSMO Multidisciplinary Analysis and Optimization Symp., Atlanta, GA, September 2002, Paper AIAA-2002-5536

[16] CHOI H.-S., KIM D.H., PARK I.H., HAHN S.Y.: 'A new design technique of magnetic systems using space mapping algorithm', *IEEE Trans. Magn.*, 2001, **37**, (5), pp. 3627–3630

doi: 10.1049/iet-map.2008.0279

[17] POWELL M.J.D.: 'Radial basis functions for multivariate interpolation: a review', in MASON J.C., COX M.G. (EDS.): 'Algorithms for approximation' (Clarendon Press, Oxford, 1987)

[18] SIMPSON T.W., MAUREY T.M., KORTE J.J., MISTREE F.: 'Kriging models for global approximation in simulation-based multidisciplinary design optimization', *AIAA J.*, 2001, **39**, (12), pp. 2233–2241

[19] SIMPSON T.W., PEPLINSKI J., KOCH P.N., ALLEN J.K.: 'Metamodels for computer-based engineering design: survey and recommendations', *Eng. Comput.*, 2001, **17**, (2), pp. 129–150

[20] QUEIPO N.V., HAFTKA R.T., SHYY W., GOEL T., VAIDYNATHAN R., TUCKER P.K.: 'Surrogate-based analysis and optimization', *Prog. Aerosp. Sci.*, 2005, **41**, (1), pp. 1–28

[21] ALEXANDROV N.M., LEWIS R.M.: 'An overview of first-order model management for engineering optimization', *Optim. Eng.*, 2001, **2**, (4), pp. 413–430

[22] 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.*, 1999, **17**, (1), pp. 1–13

[23] LEARY S.J., BHASKAR A., KEANE A.J.: 'A knowledge-based approach to response surface modeling in multifidelity optimization', *Global Optim.*, 2003, **26**, (3), pp. 297–319

[24] MORRO J.V., ESTEBAN H., BORIA V.E., BACHILLER C., COVES A.: 'New multimodel ASM technique for the efficient design of complex microwave circuits'. IEEE MTT-S Digest, Long Beach, CA, June 2005, pp. 1613–1616

[25] KOZIEL S., BANDLER J.W.: 'Controlling convergence of space-mapping algorithms for engineering optimization'. Int. Symp. Signals, Systems and Electronics, URSI ISSSE 2007, Montreal, Canada, August 2007, pp. 21–23

[26] KOZIEL S., BANDLER J.W., CHENG Q.S.: 'Adaptive space mapping with convergence enhancement for optimization of microwave structures and devices'. IEEE MTT-S Int. Microwave Symp. Dig., Atlanta, GA, June 2008, pp. 987–990

[27] KOZIEL S., BANDLER J.W., MADSEN K.: 'Quality assessment of coarse models and surrogates for space mapping optimization', *Optim. Eng.*, 2008, **9**, (4), pp. 375–391

[28] KOZIEL S., BANDLER J.W.: 'Space-mapping optimization with adaptive surrogate model', *IEEE Trans. Microw. Theory Tech.*, 2007, **55**, (3), pp. 541–547

[29] KOZIEL S., BANDLER J.W.: 'Coarse and surrogate model assessment for engineering design optimization with space mapping'. IEEE MTT-S Int. Microwave Symp. Dig., Honolulu, HI, June 2007, pp. 107–110

[30] KOZIEL S., BANDLER J.W.: 'Space mapping with multiple coarse models for optimization of microwave components', *IEEE Microw. Wirel. Compon. Lett.*, 2008, **18**, (1), pp. 1–3

[31] WU K.L., ZHAO Y.J., WANG J., CHENG M.K.K.: 'An effective dynamic coarse model for optimization design of LTCC RF circuits with aggressive space mapping', *IEEE Trans. Microw. Theory Tech.*, 2004, **52**, (1), pp. 393–402

[32] KOZIEL S., BANDLER J.W.: 'Interpolated coarse models for microwave design optimization with space mapping', *IEEE Trans. Microw. Theory Tech.*, 2007, **55**, (8), pp. 1739–1746

[33] KOZIEL S., BANDLER J.W.: 'Space mapping with distributed fine model evaluation for optimization of microwave structures and devices'. IEEE MTT-S Int. Microwave Symp. Dig., Atlanta, GA, June 2008, pp. 1377–1380

[34] KOZIEL S., CHENG Q.S., BANDLER J.W.: 'Improving efficiency of space mapping optimization of microwave structures and devices'. IEEE MTT-S Int. Microwave Symp. Dig., Honolulu, HI, June 2007, pp. 1995–1998

[35] KOZIEL S., BANDLER J.W.: 'SMF: a user-friendly software engine for space-mapping-based engineering design optimization'. Int. Symp. Signals, Systems and Electronics, URSI ISSSE 2007, Montreal, Canada, August 2007, pp. 157–160

[36] CHENG Q.S., BANDLER J.W., KOZIEL S.: 'Combining coarse and fine models for optimal design', *Microw. Mag.*, 2008, **9**, (1), pp. 79–88

[37] ZHANG L., CAO Y., WAN S., KABIR H., ZHANG Q.J.: 'Parallel automatic model generation technique for microwave modeling'. IEEE MTT-S Int. Microwave Symp. Dig., Honolulu, HI, June 2007, pp. 103–106

[38] Agilent ADS, Version 2005C, Agilent Technologies, 1400 Fountaingrove Parkway, Santa Rosa, CA 95403-1799, 2005

[39] Matlab$^{\text{TM}}$, Version 7.6, The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, 2008

[40] LEE J.R., CHO J.H., YUN S.W.: 'New compact bandpass filter using microstrip $\lambda/4$ resonators with open stub inverter', *IEEE Microw. Guided Wave Lett.*, 2000, **10**, (12), pp. 526–527

[41] FEKO$^{®}$ User's Manual, Suite 5.3, July 2007, EM Software & Systems-S.A. (Pty) Ltd, 32 Techno Lane, Technopark, Stellenbosch, 7600, South Africa, http://www.feko.info

[42] KUO J.T., CHEN S.P., JIANG M.: 'Parallel-coupled microstrip filters with over-coupled end stages for suppression of spurious responses', *IEEE Microw. Wirel. Compon. Lett.*, 2003, **13**, (10), pp. 440–442

[43] ***em***$^{\text{TM}}$ Version 11.53, Sonnet Software, Inc., 100 Elwood Davis Road, North Syracuse, NY 13212, USA, 2007

doi: 10.1049/iet-map.2008.0279