ADVANCED ELECTROMAGNETIC DESIGN OF ANTENNAS AND MICROWAVE STRUCTURES WITH SPACE MAPPING TECHNOLOGY

SOS-00-21-R

June 2000

ADVANCED ELECTROMAGNETIC DESIGN OF ANTENNAS AND MICROWAVE STRUCTURES WITH SPACE MAPPING TECHNOLOGY

Abstract

For the first time, Space Mapping (SM) technology is applied to the design of patch antennas. The aim of SM is to integrate time-consuming electromagnetic (EM) simulations into the design of antennas and microwave structures in the most efficient way. With the aid of a physically based "coarse" model, which is less accurate but fast, SM optimization technique aligns EM "fine" model responses with the "coarse" model responses automatically. The number of EM simulations involved is typically of the order of the number of optimization variables. Two typical methods, Surrogate Model based Space Mapping (SMSM) optimization and Neural Space Mapping (NSM) optimization are discussed. SMSM formulates the EM design problem into a general optimization problem. It exploits a linearized frequency sensitive mapping in the design iterations. NSM optimization is a novel technique to implement space mapping through artificial neural network (ANN). The mapping from "fine" to "coarse" model space, including frequency or any other independent variable, is constructed iteratively during optimization. These SM optimization techniques are illustrated successfully by several design problems in both microwave and antenna engineering: an H-plane resonator filter, a rectangular patch antenna and an impedance matching network for a rectangular patch antenna.

I. INTRODUCTION

EM design is being accepted more and more widely as an advantageous technique over traditional design methodology in antennas and microwave engineering, which generally employs physically based "coarse" models. However, the computationally intensive EM simulations make direct EM design, which utilizes EM "fine" model directly in the optimization loop, prohibitively time-consuming. Taking advantage of a fast "coarse" model, SM technology bridges the gap between the "coarse" model and the "fine" model.

In comparison with a full-wave EM "fine" model, the so-called "coarse" model is a rough representation of an EM system. The "coarse" model is less accurate but much faster than the "fine"

model. However, it should be capable of exhibiting the principle characteristics of the "fine" model. A coarse model can be an equivalent circuit model, an approximate analytical model or an EM model with a coarse grid. The equivalent circuit model is the circuit formulation of an EM system. The approximate analytical model can be constructed by simplifying the original structure by an approximate structure with an analytical solution. The EM "fine" model can be a Finite Element Method (FEM) [1] or a Method of a Moment (MoM) simulator [2, 3]. Both of them simulate an EM system on the basis of a discretized grid. An EM model with a coarse grid could be a "coarse" model with respect to a fine grid EM model [4].

SM technology is a further development instead of a discard of traditional design methodology. Firstly, the design parameters of a coarse model are optimized to satisfy the design specifications. With the aid of the coarse model, SM optimization technique then aligns EM "fine" model responses with the optimal "coarse" model responses automatically. The number of EM simulations involved is typically of the order of the number of optimization variables. The successful application of SM technology in microwave engineering paves a way to applying SM technology in antenna engineering.

Design methodology based on genetic algorithm (GA) and MoM has been reported in the design of a dual band patch antenna [5] and the geometry of a patch antenna [6]. MoM in spatial domain [7] is the commonly used numerical method in rigorous analysis of patch antennas. They formulate an EM boundary value problem into a system of linear equations. When forming the coefficient matrix, the numerical computation of a Sommerfeld integral is encountered, which has an oscillatory and slow-converging behavior. It is, therefore, very time-consuming. GA has been applied to engineering optimization as a global optimum searching strategy. GA is structured to solve an optimization problem by imitating the process occurring in natural evolution. It utilizes mutation operator to escape from a local minimum. It is a stochastic technique rather than a deterministic one. The penalty for the global property is low efficiency, especially in case of combining with computationally intensive MoM.

When the shape of a patch antenna is chosen and only the dimensions of the patch antenna need to be optimized, GA/MoM integration method is rather inefficient. Generally, an initial design, which is based on an empirical model (transmission line model, cavity model or multiport network model [8]), can

be achieved easily. In this case, SM technology provides a more efficient way to the EM design of patch antennas. Since SM concept has been proposed firstly [9] in EM design, many other methods on SM optimization have been developed.

Unlike the original SM optimization technique [9] which constructs a linear mapping using some fine model simulations, Aggressive Space Mapping (ASM) algorithm [4] aims at finding the optimal fine model instead of the mapping itself by exploiting quasi-Newton method. It utilizes as few fine model simulations as possible. Parameter extraction (PE) is a crucial part of ASM algorithm. It extracts a set of coarse model parameters whose response matches a given fine model response. PE is formulated into a classical optimization problem in SM. It may lead to nonunique solutions, which can make the ASM algorithm diverge. Multi-point PE [10] has been suggested as an efficient way to improve the uniqueness of PE. Trust region ASM (TRASM) algorithm [11] automates the multi-point PE in ASM. Based on TRASM algorithm, Hybrid ASM (HASM) algorithm [12] addresses the problem of a poor coarse model. The algorithm carries out SM optimization as long as SM is converging. Otherwise, it switches to direct optimization. Surrogate Model based Space Mapping (SMSM) optimization algorithm [13] draws upon recent developments in both surrogate model-based optimization and modeling of microwave devices. The surrogate model, which is a convex combination of a mapped coarse model and a linearized fine model, can be considered as an improved coarse model. It exploits a linear frequency-sensitive mapping in a novel way. This approach is shown to be especially powerful if a significant response shift exists. Neural Space Mapping (NSM) is an important development of SM technology. NSM optimization technique [14] is proposed to exploit SM-based neuromodeling techniques [15]. It implements space mapping through artificial neural network. The mapping from “fine” to “coarse” model space, including frequency or any other independent variable, is constructed iteratively during optimization.

SMSM and NSM optimization techniques are discussed in Section II and Section III, respectively. Several design problems using SM technology are illustrated in Section IV. The design of an H-plane resonator filter is carried out by employing NSM algorithm first. Then the design of a rectangular patch antenna with inset is solved by SMSM algorithm. Furthermore, SMSM algorithm is

applied to design an impedance matching network for enhancing the input impedance bandwidth of a rectangular patch antenna. The conclusions are given in Section IV.

II. SMSM OPTIMIZATION TECHNIQUE

A. Space Mapping Optimization vs. Optimization via Surrogates

We denote the fine model responses at a point $\boldsymbol{x}_f \in \Re^{n\times 1}$ and frequency ω by $\boldsymbol{R}_f(\boldsymbol{x}_f, \omega) \in \Re^{N_r \times 1}$. These responses may include the real and imaginary parts of S_{11}, etc. The response vector $\boldsymbol{R}_f(\boldsymbol{x}_f) \in \Re^{m\times 1}$ denotes the responses over all the N_ω simulation frequencies where $m=N_r N_\omega$. The original design problem is

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

where U is the objective function and $\boldsymbol{x}_f^*$ is the optimal fine model design. Solving (1) using direct optimization methods, e.g. [16], can be prohibitive due to the model's intensive simulation time.

SM optimization exploits the existence of a fast but less accurate "coarse" model of the circuit. We denote by $\boldsymbol{x}_c \in \Re^{n\times 1}$ and $\boldsymbol{R}_c(\boldsymbol{x}_c) \in \Re^{m\times 1}$ a coarse model point with corresponding response, respectively. SM establishes a mapping $\boldsymbol{x}_c = \boldsymbol{P}(\boldsymbol{x}_f)$ between the two spaces such that $\boldsymbol{R}_f(\boldsymbol{x}_f) \approx \boldsymbol{R}_c(\boldsymbol{x}_c)$ [12]. The space-mapped design $\overline{\boldsymbol{x}}_f$ is a solution of the nonlinear system

$$\boldsymbol{f}(\boldsymbol{x}_f) = \boldsymbol{P}(\boldsymbol{x}_f) - \boldsymbol{x}_c^* = 0 \tag{2}$$

where $\boldsymbol{P}(\boldsymbol{x}_f)$ is approximated through Parameter Extraction (PE) and $\boldsymbol{x}_c^*$ is the optimal coarse design.

Previous SM-based optimization algorithms solve (2) iteratively. Let $\boldsymbol{x}_f^{(i)}$ be the *i*th iterate in the solution of (2). The original ASM algorithm utilizes a quasi-Newton step. Our TRASM algorithm minimizes $\left\| \boldsymbol{f}(\boldsymbol{x}_f^{(i+1)}) \right\|$ using least squares within a trust region. The HASM algorithm addresses the

problem of a poor coarse model. It utilizes SM as long as it is converging. Otherwise, it switches to a direct optimization phase.

Here, a time-intensive model is optimized using a "surrogate" model [17]. We denote the surrogate model in the *i*th iteration by $\boldsymbol{R}_s^{(i)}(\boldsymbol{x}_f) \in \Re^{m\times 1}$. The step taken is obtained by solving

$$\boldsymbol{h}^{(i)} = arg\left\{ \min_{\boldsymbol{h}^{(i)}} \; U(\boldsymbol{R}_s^{(i)}(\boldsymbol{x}_f^{(i)} + \boldsymbol{h}^{(i)})) \right\}, \; \left\|\boldsymbol{h}^{(i)}\right\| \le \delta^{(i)} \tag{3}$$

where $\delta^{(i)}$ is the trust region size. The point $\boldsymbol{x}_f^{(i)} + \boldsymbol{h}^{(i)}$ is then validated using fine model simulation. It is accepted if it improves the desired fine model objective function. Otherwise, the accuracy of $\boldsymbol{R}_s^{(i)}(\boldsymbol{x}_f)$ should be improved using fine model validations. Additional fine model simulations may be generated to improve the surrogate model in certain directions of the parameter space.

B. Our Surrogate Model

In the *i*th iteration, the SMSM algorithm utilizes a surrogate model expressed as a convex combination between a linearized fine model (LFM) and a mapped coarse model (MCM) $\boldsymbol{R}_m^{(i)}(\boldsymbol{x}_f)$. It is given by

$$\boldsymbol{R}_s^{(i)}(\boldsymbol{x}_f) = \lambda^{(i)} \boldsymbol{R}_m^{(i)}(\boldsymbol{x}_f) + (1-\lambda^{(i)})(\boldsymbol{R}_f(\boldsymbol{x}_f^{(i)}) + \boldsymbol{J}_f^{(i)} \Delta \boldsymbol{x}_f), \; \lambda^{(i)} \in [0,1] \tag{4}$$

$\boldsymbol{J}_f^{(i)} \in \Re^{m\times n}$ is an approximation to the Jacobian of fine model responses at $\boldsymbol{x}_f^{(i)}$. The parameter $\lambda^{(i)}$ determines which of the models LFM or MCM is favored. The LFM in (4) ensures that the algorithm will work even when the coarse model is poor or even wrong.

The MCM utilizes the linear frequency-space mapping

$$\boldsymbol{R}_f(\boldsymbol{x}_f, \omega_j) \approx \boldsymbol{R}_m^{(i)}(\boldsymbol{x}_f, \omega_j) = \boldsymbol{R}_c(\boldsymbol{P}^{(i)}(\boldsymbol{x}_f, \omega_j), P_\omega^{(i)}(\boldsymbol{x}_f, \omega_j)), \; j=1, 2, \ldots, N_\omega \tag{5}$$

where

$$\begin{bmatrix} \boldsymbol{P}^{(i)}(\boldsymbol{x}_f, \omega_j) \\ P_\omega^{(i)}(\boldsymbol{x}_f, \omega_j) \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}^{(i)} & \boldsymbol{s}^{(i)} \\ \boldsymbol{t}^{(i)T} & \sigma^{(i)} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{x}_f \\ \omega_j \end{bmatrix} + \begin{bmatrix} \boldsymbol{c}^{(i)} \\ \gamma^{(i)} \end{bmatrix}, \tag{6}$$

and $\Delta \boldsymbol{x}_f = \boldsymbol{x}_f - \boldsymbol{x}_f^{(i)}$. The parameters $\boldsymbol{B}^{(i)} \in \Re^{n\times n}$, $\boldsymbol{s}^{(i)} \in \Re^{n\times 1}$, $\boldsymbol{t}^{(i)} \in \Re^{n\times 1}$, $\boldsymbol{c}^{(i)} \in \Re^{n\times 1}$, $\sigma^{(i)} \in \Re^{1\times 1}$ and $\gamma^{(i)} \in \Re^{1\times 1}$ are the mapping parameters. ω_j is the *j*th simulation frequency, *j*=1, 2, …, N_ω. Here, a fine model point $\boldsymbol{x}_f$ and frequency ω_j correspond to a coarse point $\boldsymbol{P}^{(i)}(\boldsymbol{x}_f, \omega_j)$ and frequency $P_\omega^{(i)}(\boldsymbol{x}_f, \omega)$.

The advantage of utilizing the frequency-sensitive mapping (6) is illustrated for a single response case by Fig. 1. It is required to extract the coarse point $\boldsymbol{x}_c$ corresponding to a given fine point $\boldsymbol{x}_f$. The PE optimizer may not have enough information to align the almost disjoint responses. However, the responses could align perfectly if a frequency mapping $\omega_c = P_\omega(\omega)$ is applied to the coarse frequency axis. Fig. 1(b) illustrates possible aligned results.

The mapping parameters of (6) are obtained so that the MCM approximates the fine model over a region of fine model parameters and frequency. They are obtained by the optimization procedure

$$[\boldsymbol{B}^{(i)}, \boldsymbol{s}^{(i)}, \boldsymbol{t}^{(i)}, \sigma^{(i)}, \boldsymbol{c}^{(i)}, \gamma^{(i)}] = arg\left\{ \min_{\boldsymbol{B}, \boldsymbol{s}, \boldsymbol{t}, \sigma, \boldsymbol{c}, \gamma} \left\| \begin{bmatrix} \boldsymbol{e}_1^T & \boldsymbol{e}_2^T & \cdots & \boldsymbol{e}_N^T \end{bmatrix}^T \right\| \right\} \tag{7}$$

$$\boldsymbol{e}_j = \boldsymbol{R}_c(\boldsymbol{P}^{(i)}(\boldsymbol{x}_f^{(j)}, \omega_k), P_\omega^{(i)}(\boldsymbol{x}_f^{(j)}, \omega_k)) - \boldsymbol{R}_f(\boldsymbol{x}_f^{(j)}, \omega_k) \;\forall\, \boldsymbol{x}_f^{(j)} \in V^{(i)},\; \forall \omega_k, k = 1, 2, \ldots, N_\omega,\; N = N_p N_\omega \tag{8}$$

where $V^{(i)}$ is a set of fine model points whose cardinality is $\left|V^{(i)}\right| = N_p$. $V^{(i)}$ is mainly composed of a subset of previously simulated points. It contains points that are within an α-neighborhood of $\boldsymbol{x}_f^{(i)}$ and sufficiently cover this neighborhood. Additional points may be simulated by the algorithm. This occurs if the algorithm fails to make a successful iteration and the previously simulated fine points do not adequately cover the α-neighborhood.

C. The SMSM Algorithm

The *i*th iteration of the algorithm proceeds as follows. First, the set $V^{(i)}$ is constructed. The mapping parameters are then estimated using the optimization procedure (7)-(8). The step $\boldsymbol{h}^{(i)}$ is

obtained by solving (3), where the surrogate model is given by (4). Notice that (3) utilizes only coarse model simulations and can be solved by traditional optimization methods.

$\boldsymbol{h}^{(i)}$ is accepted if it improves the objective function. Otherwise, it is rejected. The parameters $\boldsymbol{J}_f^{(i)}$, $\delta^{(i)}$ and $\lambda^{(i)}$ are updated in each iteration. Broyden's formula [18] is used to update $\boldsymbol{J}_f^{(i)}$. Initially, we set $\boldsymbol{J}_f^{(1)} = \boldsymbol{J}_c^*$, the Jacobian of the coarse model response at $\boldsymbol{x}_c^*$. The trust region $\delta^{(i)}$ is updated based on how the actual reduction r_a in U matches the predicted reduction r_p. The ratio

$$\rho = \frac{r_a}{r_p} = \frac{U(\boldsymbol{R}_f(\boldsymbol{x}_f^{(i)})) - U(\boldsymbol{R}_f(\boldsymbol{x}_f^{(i)} + \boldsymbol{h}^{(i)}))}{U(\boldsymbol{R}_s^{(i)}(\boldsymbol{x}_f^{(i)})) - U(\boldsymbol{R}_s^{(i)}(\boldsymbol{x}_f^{(i)} + \boldsymbol{h}^{(i)}))} \tag{9}$$

is thus evaluated at the end of each iteration. If $\rho \geq 0.75$, the surrogate model has good accuracy and we set $\delta^{(i+1)} = \pi_1 \delta^{(i)}$, $\pi_1 > 1.0$. If $\rho \leq 0.10$, we set $\delta^{(i+1)} = \pi_2 \delta^{(i)}$, $0 < \pi_2 < 1.0$. Otherwise, we set $\delta^{(i+1)} = \delta^{(i)}$. $\lambda^{(i)}$ is updated to favor the more accurate model, either the LFM or the MCM. It is initialized by $\lambda^{(1)} = 1$. The actual update utilized is

$$\lambda^{(i+1)} = \frac{\left\| \boldsymbol{E}_l^{(i)} \right\|}{\left\| \boldsymbol{E}_l^{(i)} \right\| + \left\| \boldsymbol{E}_m^{(i)} \right\|} \tag{10}$$

where $\boldsymbol{E}_m^{(i)} = \boldsymbol{R}_m^{(i)}(\boldsymbol{x}_f^{(i)} + \boldsymbol{h}^{(i)}) - \boldsymbol{R}_f(\boldsymbol{x}_f^{(i)} + \boldsymbol{h}^{(i)})$ and $\boldsymbol{E}_l^{(i)} = \boldsymbol{R}_f(\boldsymbol{x}_f^{(i)}) + \boldsymbol{J}_f^{(i)} \boldsymbol{h}^{(i)} - \boldsymbol{R}_f(\boldsymbol{x}_f^{(i)} + \boldsymbol{h}^{(i)})$ define the prediction error using the MCM and the LFM, respectively. The AFSM algorithm terminates if n+1 consecutive unsuccessful iterations are carried out or if $\left\| \boldsymbol{h}^{(i)} \right\|$ becomes sufficiently small. Fig. 2 illustrates one iteration.

The algorithm can be summarized by the following steps

Step 1. Given $\boldsymbol{x}_f^{(1)} = \boldsymbol{x}_c^*$, $\lambda^{(1)} = 1$, $\delta^{(1)}$, α, $\boldsymbol{J}_f^{(1)} = \boldsymbol{J}_c^*$ and i=1.

Step 2. Construct $V^{(i)}$.

Step 3. Apply the optimization procedure (7)-(8) to obtain the mapping parameters.

Step 4. Obtain the suggested step $\boldsymbol{h}^{(i)}$ by solving (3).

Step 5. If $U(\boldsymbol{R}_f(\boldsymbol{x}_f^{(i)} + \boldsymbol{h}^{(i)})) < U(\boldsymbol{R}_f(\boldsymbol{x}_f^{(i)}))$, set $\boldsymbol{x}_f^{(i+1)} = \boldsymbol{x}_f^{(i)} + \boldsymbol{h}^{(i)}$ else $\boldsymbol{x}_f^{(i+1)} = \boldsymbol{x}_f^{(i)}$.

Step 6. Update $\boldsymbol{J}_f^{(i)}$, $\delta^{(i)}$ and $\lambda^{(i)}$.

Step 7. If the stopping criterion is satisfied stop.

Step 8. Set i=i+1 and go to Step 2.

III. NSM OPTIMIZATION TECHNIQUE

A. NSM Optimization: An Overview

We start by finding the optimal solution $\boldsymbol{x}_c^*$ that yields the desired response using the coarse model. We select 2n additional points following an n-dimensional star distribution [15] centered at $\boldsymbol{x}_c^*$, where n is the number of design parameters ($\boldsymbol{x}_c$, $\boldsymbol{x}_f$). The percentage of deviation from $\boldsymbol{x}_c^*$ for each design parameter is determined according to the coarse model sensitivity. The larger the sensitivity of the coarse model response w.r.t. a certain parameter, the smaller the percentage of variation of that parameter. We assume that the coarse model sensitivity is similar to that one of the fine model.

The fine model response $\boldsymbol{R}_f$ at the optimal coarse solution $\boldsymbol{x}_c^*$ is then calculated. If $\boldsymbol{R}_f$ is approxi-mately equal to the desired response, the algo-rithm ends, otherwise we develop an SM-based neuromodel over the 2n+1 fine model points.

Once an SM-based neuromodel with small learning errors is available, *we use it as an improved coarse model*, optimizing its parameters to generate the desired response. The solution to this problem becomes the next point in the fine model parameter space, and it is included in the learning set.

We calculate the fine model response at the new point, and compare it with the desired response. If it is still different, we re-train the SM-based neuromodel over the extended set of learning samples and the algorithm continues. If not, the algorithm terminates.

B. Coarse Optimization

During the coarse optimization phase of NFSM optimization, we want to find the optimal coarse model solution $\boldsymbol{x}_c^*$ that generates the desired response over the frequency range of interest. The vector of coarse model responses $\boldsymbol{R}_c$ might contain N_r different responses (for example, $|S_{11}|$ and $|S_{21}|$),

$$\boldsymbol{R}_c(\boldsymbol{x}_c) = [\boldsymbol{R}_c^1(\boldsymbol{x}_c)^T \quad \ldots \quad \boldsymbol{R}_c^{N_r}(\boldsymbol{x}_c)^T]^T \tag{11}$$

where each individual response has been sampled at N_ω frequency points,

$$\boldsymbol{R}_c^r(\boldsymbol{x}_c) = [R_c^r(\boldsymbol{x}_c, \omega_1) \quad \ldots \quad R_c^r(\boldsymbol{x}_c, \omega_{N_\omega})]^T \quad r = 1, \ldots, N_\omega \tag{12}$$

It notices that $\boldsymbol{R}_c(\boldsymbol{x}_c)$ in this section is the same as it in Section II. The desired response $\boldsymbol{R}^*$ is expressed in terms of specifications. The problem of circuit design using the coarse model can be formulated as [19]

$$\boldsymbol{x}_c^* = arg \left\{ \min_{\boldsymbol{x}_c} U(\boldsymbol{R}_c(\boldsymbol{x}_c)) \right\} \tag{13}$$

where U is a suitable objective function. For example, U could be a minimax objective func-tion expressed in terms of upper and lower specifications for each response and frequency sample. A rich collection of objective functions, for different design constraints, is in [19].

C. Training The SM-Based Neuromodel During NSM Optimization

At the *i*th iteration, we find the simplest neuro-mapping $\boldsymbol{P}^{(i)}$ such that the coarse model using that mapping approximates the fine model at all the learning points. It should be noted that $\boldsymbol{P}^{(i)}$ in this section is different from that in Section II. This is realized by solving

$$\boldsymbol{w}^* = arg \left\{ \min_{\boldsymbol{w}} \left\| [\cdots \quad \boldsymbol{e}_s^T \quad \cdots]^T \right\| \right\} \tag{14}$$

with

$$\boldsymbol{e}_s = \boldsymbol{R}_f(\boldsymbol{x}_f^{(l)}, \omega_j) - \boldsymbol{R}_c(\boldsymbol{x}_{c_j}^{(l)}, \omega_{c_j}) \tag{15a}$$

$$\begin{bmatrix} \boldsymbol{x}_{c_j}^{(l)} \\ \omega_{c_j} \end{bmatrix} = \boldsymbol{P}^{(i)}(\boldsymbol{x}_f^{(l)}, \omega_j, \boldsymbol{w}) \tag{15b}$$

$$j = 1, \ldots, N_\omega, \; l = 1, \ldots, 2n+i, \; s = j + N_\omega(l-1) \tag{15c}$$

where $2n + i$ is the number of training base points and N_ω is the number of frequency points per frequency

sweep. The total number of learning samples at the *i*th iteration is $s = (2n + i)\, N_\omega$.

(15b) is the input-output relationship of the ANN that implements the mapping at the *i*th iteration. Vector $\boldsymbol{w}$ contains the internal parameters (weights, bias, etc.) of the ANN. The paradigm chosen to implement $\boldsymbol{P}^{(i)}$ is a 3-layer perceptron.

All the SM-based neuromodeling techniques proposed in [15] can be exploited to solve (14). The starting point for the first training is a unit mapping, i.e., $\boldsymbol{P}^{(0)}(\boldsymbol{x}_f^{(l)}, \omega_j, \boldsymbol{w}_u) = [\boldsymbol{x}_f^{(l)T}\ \omega_j]^T$, for $j = 1,\ldots, N_\omega$ and $l = 1,\ldots, 2n+1$, where $\boldsymbol{w}_u$ contains the internal parameters of the ANN for a unit mapping. The SM-based neuromodel is trained in the next iterations using the previous mapping as the starting point.

The complexity of the ANN is gradually in-creased according to the learning error ε_L, starting with a linear mapping (3-layer percep-tron with 0 hidden neurons). In other words, we use the simplest ANN that yields an acceptable learning error, defined as

$$\varepsilon_L = \left\| [\cdots \quad \boldsymbol{e}_s^{\,T} \quad \cdots]^T \right\| \tag{16}$$

where $\boldsymbol{e}_s$ is obtained from (15) using the current optimal values for the ANN free parameters $\boldsymbol{w}^*$.

D. SM-Based Neuromodel Optimization

At the *i*th iteration of NSM optimization, we use an SM-based neuromodel with small learning error as an improved coarse model, optimizing its parameters to generate the desired response. We denote the SM-based neuromodel response as $\boldsymbol{R}_{SMBN}$, defined as

$$\boldsymbol{R}_{SMBN}(\boldsymbol{x}_f) = [\boldsymbol{R}^1_{SMBN}(\boldsymbol{x}_f)^T \quad \ldots \quad \boldsymbol{R}^{N_r}_{SMBN}(\boldsymbol{x}_f)^T]^T \tag{17}$$

where

$$\boldsymbol{R}^r_{SMBN}(\boldsymbol{x}_f) = [R^r_c(\boldsymbol{x}_{c1}, \omega_{c1}) \quad \ldots \quad R^r_c(\boldsymbol{x}_{cF_p}, \omega_{cF_p})]^T,\ r = 1,\ldots, N_\omega \tag{18}$$

with

$$\begin{bmatrix} \boldsymbol{x}_{c_j} \\ \omega_{c_j} \end{bmatrix} = \boldsymbol{P}^{(i)}(\boldsymbol{x}_f, \omega_j, \boldsymbol{w}^*) \tag{19}$$

and *j* defined in (15c). The solution to the following optimization problem becomes the next iterate:

$$\boldsymbol{x}_f^{(2n+i+1)} = arg \left\{ \min_{\boldsymbol{x}_f} U(\boldsymbol{R}_{SMBN}(\boldsymbol{x}_f)) \right\} \tag{20}$$

If an SMN neuromapping is used to implement $\boldsymbol{P}^{(i)}$, the next iterate can be obtained in a simpler manner

$$\boldsymbol{x}_f^{(2n+i+1)} = arg \left\{ \min_{\boldsymbol{x}_f} \left\| \boldsymbol{P}_{SM}^{(i)}(\boldsymbol{x}_f, \boldsymbol{w}^*) - \boldsymbol{x}_c^* \right\| \right\} \tag{21}$$

E. NSM Algorithm

Step 1. Find $\boldsymbol{x}_c^*$ by solving (13).

Step 2. Choose $\boldsymbol{x}_f^{(1)}, \ldots, \boldsymbol{x}_f^{(2n)}$ following a star distribution around $\boldsymbol{x}_c^*$.

Step 3. Initialize $i = 1$, $\boldsymbol{x}_f^{(2n+i)} = \boldsymbol{x}_c^*$.

Step 4. Stop if $\left\| \boldsymbol{R}_f(\boldsymbol{x}_f^{(2n+i)}, \omega_j) - \boldsymbol{R}_c(\boldsymbol{x}_c^*, \omega_j) \right\| \le \varepsilon_R$, $j = 1, \ldots, N_\omega$.

Step 5. Initialize $\boldsymbol{P}^{(i)} = \boldsymbol{P}^{(i-1)}$, where $\boldsymbol{P}^{(0)}(\boldsymbol{x}_f^{(l)}, \omega_j, \boldsymbol{w}_u) = \begin{bmatrix} \boldsymbol{x}_f^{(l)} \\ \omega_j \end{bmatrix}$, $j = 1, \ldots, N_\omega$; $l = 1, \ldots, 2n+i$.

Step 6. Find $\boldsymbol{w}^*$ by solving (14).

Step 7. Calculate ε_L using (16).

Step 8. If $\varepsilon_L > \varepsilon_{\min}$, increase the complexity of $\boldsymbol{P}^{(i)}$ and go to Step 6.

Step 9. If an SM neuromapping is used to implement $\boldsymbol{P}^{(i)}$, solve (11), otherwise solve (20).

Step 10. Set $i = i + 1$; go to Step 4.

IV. EXAMPLES

A. An H-plane Resonator Filter.

B. A Rectangular Patch Antenna.

C. An Impedance Matching Network for A Rectangular Patch Antenna.

V. CONCLUSIONS

REFERENCES

[1] Agilent HFSS™, Version 5.5, Agilent Technologies, 1400 Fountaingrove Parkway, Santa Rosa, CA 95403-1799.

[2] ***em***™ , Sonnet Software, Inc., 1020 Seventh North Street, Suite 210, Liverpool, NY 13088, 1997.

[3] Momentum™ Version 3.0, Agilent Technologies, 1400 Fountaingrove Parkway, Santa Rosa, CA 95403-1799, 1998.

[4] J.W. Bandler, R.M. Biernacki, S.H. Chen, R.H. Hemmers and K. Madsen, "Electromagnetic optimization exploiting aggressive space mapping," *IEEE Trans. Microwave Theory Tech.*, vol. 43, 1995, pp. 2874-2882.

[5] J.M. Johnson and Y. Rahmat-Samii, "Genetic algorithms and Method of Moments (GA/MoM) for the design of integrated antennas," *IEEE Trans. Antennas Propagation*, vol. 47, 1999, pp. 1606-1614.

[6] L. Alatan, M.I. Aksun, K. Leblebicioğlu and M.T. Birand, "Use of computionally efficient Method of Moments in the optimization of printed antennas," *IEEE Trans. Antennas Propagation*, vol. 47, 1999, pp. 725-732.

[7] L. Alatan, M.I. Aksun, K. Mahadevan and M.T. Birand, "Analytical evaluation of the MoM matrix elements," *IEEE Trans. Antennas Propagation*, vol. 44, 1996, pp. 519-525.

[8] D.M. Pozar and D.H. Schubert, *Microstrip Antennas*: *The Analysis and Design of Microstrip Antennas and Arrays*. New York: IEEE Press, 1995.

[9] J.W. Bandler, R.M. Biernacki, S.H. Chen, P.A. Grobelny and R.H. Hemmers, "Space mapping technique for electromagnetic optimization," *IEEE Trans. Microwave Theory Tech.*, vol. 42, 1994, pp. 2536-2544.

[10] J.W. Bandler, R.M. Biernacki and S.H. Chen, "Fully automated space mapping optimization of 3D structures," *IEEE MTT-S Int. Microwave Symp. Dig*. (San Francisco, MA), 1996, pp. 753-756.

[11] M.H. Bakr, J.W. Bandler, R.M. Biernacki, S.H. Chen and K. Madsen, "A trust region aggressive space mapping algorithm for EM optimization," *IEEE Trans. Microwave Theory Tech.*, vol. 46, 1998, pp. 2412-2425.

[12] M.H. Bakr, J.W. Bandler, N. Georgieva and K. Madsen, "A hybrid aggressive space mapping algorithm for EM optimization," *IEEE Trans. Microwave Theory Tech.*, vol. 47, 1999, pp. 2440-2449.

[13] M.H. Bakr, J.W. Bandler, K. Madsen, J.E. Rayas-Sánchez and J. Søndergaard, "Surrogate model based space mapping optimization of microwave circuits," *IEEE MTT-S Int. Microwave Symp. Dig*. (Boston, MA), 2000.

[14] M.H. Bakr, J.W. Bandler, M.A. Ismail, J.E. Rayas-Sánchez and Q.J. Zhang, "Neural space mapping EM optimization of microwave structures," *IEEE MTT-S Int. Microwave Symp. Dig*. (Boston, MA), 2000.

[15] J.W. Bandler, M.A. Ismail, J.E. Rayas-Sánchez and Q.J. Zhang, "Neuromodeling of microwave circuits exploiting space mapping technology," *IEEE Trans. Microwave Theory Tech.*, vol. 47, 1999, pp. 2417-2427.

[16] J.W. Bandler, W. Kellermann and K. Madsen, "A superlinearly convergent minimax algorithm for microwave circuit design," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-33, 1985, pp. 1519-1530.

[17] A.J. Booker, J.E. Dennis, Jr., P.D. Frank, D. B. Serafini, V. Torczon and M.W. Trosset, "A rigorous framework for optimization of expensive functions by surrogates," *Structural Optimization*, vol. 17, 1999, pp. 1-13.

[18] C.G. Broyden, "A class of methods for solving nonlinear simultaneous equations," *Math. Comp.*, vol. 19, 1965, pp. 577-593.

[19] J.W. Bandler and S.H. Chen, "Circuit optimization: the state of the art," *IEEE Trans. Microwave Theory Tech.*, vol. 36, 1988, pp. 424-443.

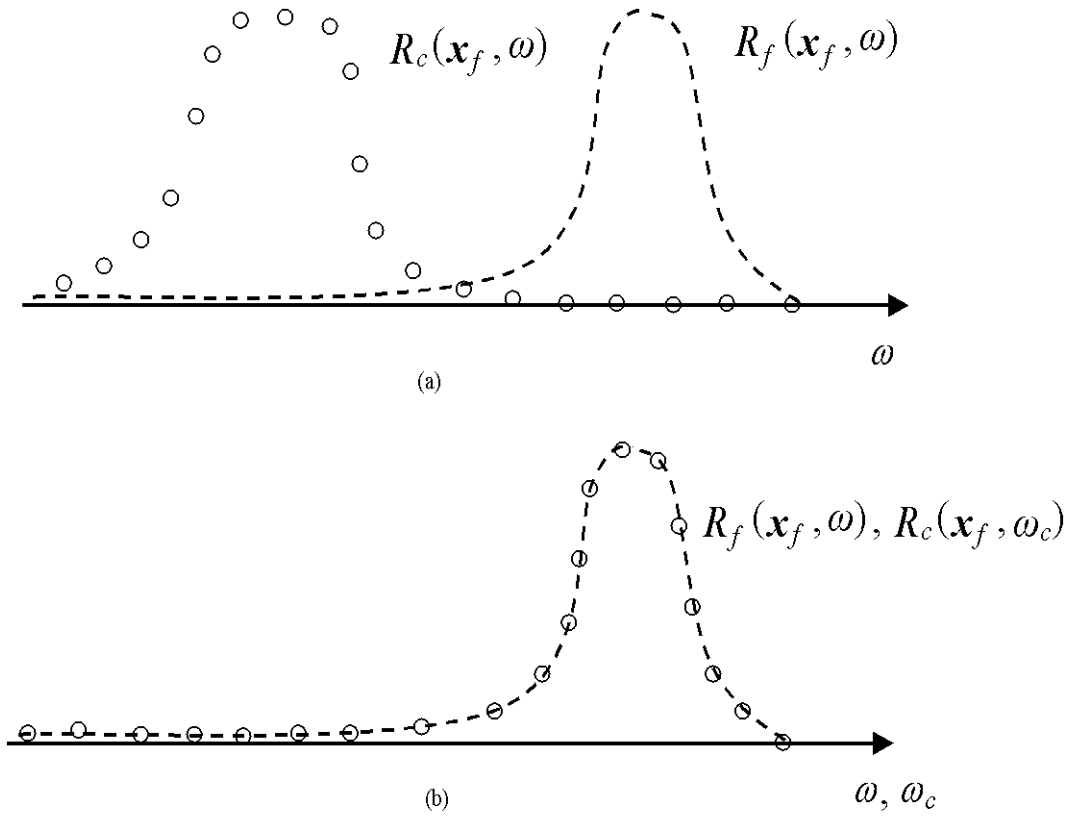

Fig. 1. Illustration of the frequency-sensitive mapping concept, (a) a significant frequency band shift exists between fine and coarse model responses at the initial iteration and (b) the coarse model frequency is transformed such that both responses match.

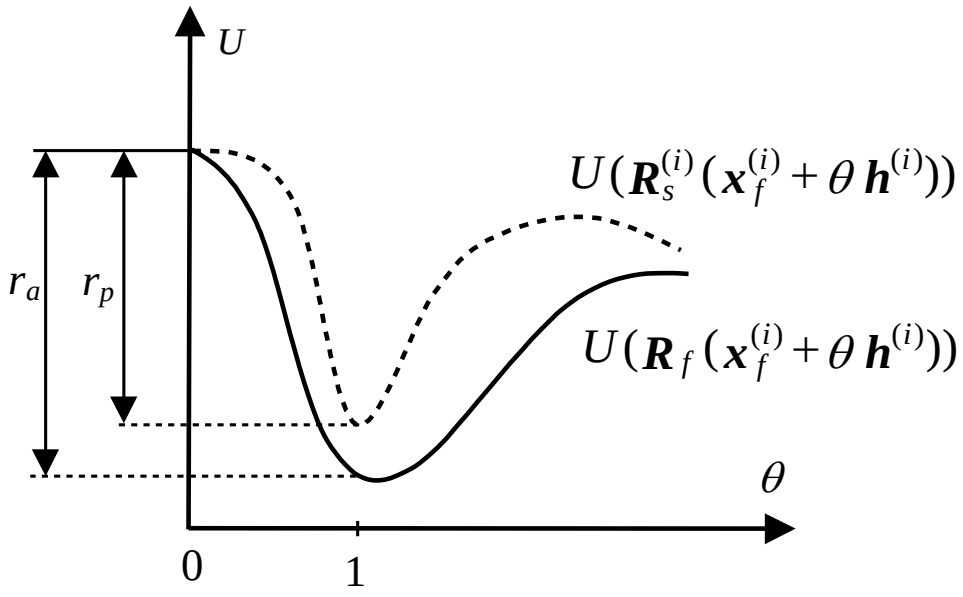

Fig. 2. Illustration of the *i*th iteration of the SMSM algorithm.