NEURAL INVERSE SPACE MAPPING (NISM) OPTIMIZATION FOR EM-BASED DESIGN OF MICROWAVE STRUCTURES

John W. Bandler, *Fellow*, *IEEE,* Mostafa A. Ismail, *Student Member*, *IEEE*,
José E. Rayas-Sánchez, *Senior Member*, *IEEE* and Qi-Jun Zhang, *Senior Member*, *IEEE*

Keywords neural network applications, space mapping, optimization methods, design automation, EM optimization, neural space mapping, microwave circuits, microstrip filters, neural modeling

Abstract We present Neural Inverse Space Mapping (NISM) optimization for electromagnetics-based design of microwave structures. The inverse of the mapping from the fine to the coarse model parameter spaces is exploited for the first time in a space mapping algorithm. NISM optimization does not require up-front EM simulations, multipoint parameter extraction or frequency mapping. It employs a simple statistical parameter extraction procedure. The inverse of the mapping is approximated by a neural network whose generalization performance is controlled through a network growing strategy. We contrast our new algorithm with Neural Space Mapping (NSM) optimization as well as with our Trust Region Aggressive Space Mapping exploiting Surrogates.

I. INTRODUCTION

Neural networks have been extensively used for modeling microwave devices and circuits, in many different ways [1, 2]. In contrast, the use of neural networks for design by optimization is at an earlier stage: a few variations in the use of neural networks for optimization of microwave circuits have

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grants OGP0007239 and STR234854-00, through the Micronet Network of Centres of Excellence and Bandler Corporation. J.E. Rayas-Sánchez is supported by an Ontario Graduate Scholarship, as well as by ITESO (Instituto Tecnológico y de Estudios Superiores de Occidente, Mexico). M.A. Ismail is supported by a Nortel Networks Ontario Graduate Scholarship in Science and Technology.

J.W. Bandler, M.A. Ismail and J.E. Rayas-Sánchez are with the Simulation Optimization Systems Research Laboratory and the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, Canada L8S 4K1.

J.W. Bandler is also with Bandler Corporation, P.O. Box 8083, Dundas, Ontario, Canada L9H 5E7.

Q.J. Zhang is with the Department of Electronics, Carleton University, 1125 Colonel By Drive, Ottawa, Canada K1S 5B6.

been reported. The most widely used strategy for neural optimization of microwave circuits consists of generating a neuromodel of the microwave circuit within a certain training region of the design parameters, and then applying conventional optimization to the neuromodel to find the optimal solution that yields the desired response. Full wave EM simulations are typically employed to generate the training data. The generalization ability of the neuromodel is controlled during the training process by using validation data and testing data, also obtained from EM simulations. Examples of this neural optimization approach can be found in [3-7].

Both limitations of the conventional neural optimization approach have been alleviated by incorporating prior knowledge into the neural network training scheme, following an EM-ANN approach [8, 9], or a neural space mapping (NSM) approach [10, 11]. NSM optimization also avoids the use of the validation and testing data typically needed during training.

An elegant new algorithm for EM-based design of microwave circuits is presented in this work: Neural Inverse Space Mapping (NISM) optimization [12]. This is the first Space Mapping (SM) algorithm that explicitly makes use of the inverse of the mapping from the fine to the coarse model parameter spaces. NISM follows an aggressive formulation by not requiring a number of up-front fine model evaluations to start building the mapping. A simple procedure for parameter extraction avoids the need for multipoint matching and frequency mappings.

A neural network whose generalization performance is controlled through a network growing strategy approximates the inverse of the mapping at each iteration. The NISM step consists simply of evaluating the current neural network at the optimal coarse solution. We prove that this step is equivalent to a quasi-Newton step while the inverse mapping remains essentially linear, and gradually departs from a quasi-Newton step as the amount of nonlinearity in the inverse mapping increases.

We compare our new algorithm with Neural Space Mapping (NSM) optimization [11] by solving the same microwave design problems: a bandstop microstrip filter with open stubs and an HTS microstrip filter. We also contrast NISM optimization with Trust Region Aggressive Space Mapping exploiting Surrogates.

II. NEURAL INVERSE SPACE MAPPING (NISM)

A. Notation

Let the vectors $\boldsymbol{x}_c$ and $\boldsymbol{x}_f$ represent the design parameters of the coarse and fine models, respectively ($\boldsymbol{x}_c, \boldsymbol{x}_f \in \Re^n$). We denote the optimizable fine model responses at point $\boldsymbol{x}_f$ and frequency ω by $\boldsymbol{R}_f(\boldsymbol{x}_f, \omega) \in \Re^r$ where r is the number of responses to be optimized. For example, if the responses to be optimized are $|S_{11}|$ and $|S_{21}|$, then $r = 2$. The vector $\boldsymbol{R}_f(\boldsymbol{x}_f) \in \Re^m$ denotes the fine model responses at the F_p sample frequency points, where $m = rF_p$. Similarly, $\boldsymbol{R}_c(\boldsymbol{x}_c) \in \Re^m$ denotes the corresponding coarse model responses to be optimized.

Additionally, we denote the characterizing fine model responses at point $\boldsymbol{x}_f \in \Re^n$ and frequency ω by $\boldsymbol{R}_{fs}(\boldsymbol{x}_f, \omega) \in \Re^R$, which includes the real and imaginary parts of all the available characterizing responses in the model (considering symmetry). For example, for a 2-port reciprocal network they include $\mathrm{Re}\{S_{11}\}$, $\mathrm{Im}\{S_{11}\}$, $\mathrm{Re}\{S_{21}\}$ and $\mathrm{Im}\{S_{21}\}$, and $R = 4$. The vector $\boldsymbol{R}_{fs}(\boldsymbol{x}_f) \in \Re^M$ denotes the characterizing fine model responses at all the F_p frequency points, where $M = RF_p$. Similarly, $\boldsymbol{R}_{cs}(\boldsymbol{x}_c) \in \Re^M$ denotes the corresponding characterizing coarse model responses.

B. Flow Diagram: an Overview

A flow diagram for NISM optimization is shown in Fig. 1. We start by performing regular minimax optimization on the coarse model to find the optimal coarse solution $\boldsymbol{x}_c^*$ that yields the desired response. The characterizing fine model responses $\boldsymbol{R}_{fs}$ at the optimal coarse solution $\boldsymbol{x}_c^*$ are then calculated.

We realize parameter extraction, which consists of finding the coarse model parameters that makes the characterizing coarse responses $\boldsymbol{R}_{cs}$ as close as possible to the previously calculated $\boldsymbol{R}_{fs}$.

We continue by training the simplest neural network $\boldsymbol{N}$ that implements the inverse of the mapping from the fine to the coarse parameter space at the available points.

The new point in the fine model parameter space is then calculated by simply evaluating the neural network at the optimal coarse solution. If the maximum relative change in the fine model

parameters is smaller than a previously defined amount we finish, otherwise we calculate the characterizing fine model responses at the new point and continue with the algorithm.

C. Parameter Extraction

The parameter extraction procedure at the *i*th NISM iteration is formulated as the following optimization problem

$$\boldsymbol{x}_c^{(i)} = \arg\min_{\boldsymbol{x}_c} U_{PE}(\boldsymbol{x}_c) \tag{1a}$$

$$U_{PE}(\boldsymbol{x}_c) = \left\| \boldsymbol{e}(\boldsymbol{x}_c) \right\|_2^2 \tag{1b}$$

$$\boldsymbol{e}(\boldsymbol{x}_c) = \boldsymbol{R}_{fs}(\boldsymbol{x}_f^{(i)}) - \boldsymbol{R}_{cs}(\boldsymbol{x}_c) \tag{1c}$$

We solve (1) using the Levenberg-Marquardt algorithm for nonlinear curve fitting available in the Matlab™ Optimization Toolbox [13].

We normally use $\boldsymbol{x}_c^*$ as the starting point for solving (1). This might not be a good starting point when an extremely severe matching problem is being solved, one that has some poor local minimum around $\boldsymbol{x}_c^*$. If the algorithm is trapped in a poor local minimum, we change the starting point for (1) by taking a small random perturbation $\Delta\boldsymbol{x}$ around $\boldsymbol{x}_c^*$ until we find an acceptable local minimum, i.e., until we obtain a good matching between both fine and coarse models.

The maximum perturbation $\Delta_{\max}$ is obtained from the maximum absolute sensitivity of the parameter extraction objective function at $\boldsymbol{x}_c^*$ as follows

$$\Delta_{\max} = \frac{\delta_{PE}}{\left\| \nabla U_{PE}(\boldsymbol{x}_c^*) \right\|_\infty} \tag{2}$$

Let $\boldsymbol{rand} \in \Re^n$ be a vector whose elements take random values between 0 and +1 every time it is evaluated. The values of the elements of $\Delta\boldsymbol{x}$ are calculated as

$$\Delta x_k = \Delta_{\max}(2rand_k - 1), \quad k = 1, \ldots, n \tag{3}$$

A value of $\delta_{PE} = 0.03$ is used in our implementation. Many other values of δ_{PE} could be used in (2), since we use it only to escape from a poor local minimum.

A similar strategy for statistical parameter extraction was proposed in [14], where an exploration region is first created by predefining a fixed number of starting points around $\boldsymbol{x}_c^*$.

The algorithm for realizing parameter extraction is stated as follows

Algorithm: Parameter Extraction
begin solve (1) using $\boldsymbol{x}_c^*$ as starting point while $\|\boldsymbol{e}(\boldsymbol{x}_c^{(i)})\|_\infty > \varepsilon_{PE}$ calculate $\Delta\boldsymbol{x}$ using (2) and (3) solve (1) using $\boldsymbol{x}_c^*+\Delta\boldsymbol{x}$ as starting point end

A value of $\varepsilon_{PE} = 0.15$ is used in our implementation, assuming that all the response values are normalized.

D. Inverse Neuromapping

When training the neural network $\boldsymbol{N}$ that implements the inverse mapping we solve the following optimization problem

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} U_N(\boldsymbol{w}) \tag{4a}$$

$$U_N(\boldsymbol{w}) = \left\| [\cdots \quad \boldsymbol{e}_l^T \quad \cdots]^T \right\|_2^2 \tag{4b}$$

$$\boldsymbol{e}_l = \boldsymbol{x}_f^{(l)} - \boldsymbol{N}(\boldsymbol{x}_c^{(l)}, \boldsymbol{w}), \quad l = 1,\ldots,i \tag{4c}$$

where i is the current NISM iteration and vector $\boldsymbol{w}$ contains the internal parameters (weights, bias, etc.) of the neural network $\boldsymbol{N}$. The starting point $\boldsymbol{w}^{(0)}$ for solving (4) is a unit mapping, i.e. $\boldsymbol{N}(\boldsymbol{x}_c^{(l)}, \boldsymbol{w}^{(0)}) = \boldsymbol{x}_c^{(l)}$, for $l = 1,\ldots, i$. We use the Scaled Conjugate Gradient (SCG) algorithm available in the Matlab™ Neural Network Toolbox [15] for solving (4). Notice that the time consumed in solving (4) is almost neglectable since no coarse or fine model simulations are needed.

To control the generalization performance of the neural network $\boldsymbol{N}$, we follow a network growing strategy [16], in which case we start with a small perceptron to match the initial points and then add more neurons only when we are unable to meet a small error.

We initially assume a 2-layer perceptron given by

$$\boldsymbol{N}(\boldsymbol{x}_c, \boldsymbol{w}) = \boldsymbol{x}_f = \boldsymbol{W}^o \boldsymbol{x}_c + \boldsymbol{b}^o \tag{5}$$

where $\boldsymbol{W}^o \in \Re^{n \times n}$ is the matrix of output weighting factors, $\boldsymbol{b}^o \in \Re^n$ is the vector of output bias elements, and vector $\boldsymbol{w}$ contains $\boldsymbol{b}^o$ and the columns of $\boldsymbol{W}^o$. The starting point is obtained by making $\boldsymbol{W}^o = \boldsymbol{I}$ and $\boldsymbol{b}^o = \boldsymbol{0}$.

If a 2-layer perceptron is not sufficient to make the learning error $U_N(\boldsymbol{w}^*)$ small enough, then we use a 3-layer perceptron with h hidden neurons given by

$$\boldsymbol{N}(\boldsymbol{x}_c, \boldsymbol{w}) = \boldsymbol{W}^o \boldsymbol{\Phi}(\boldsymbol{x}_c) + \boldsymbol{b}^o \tag{6a}$$

$$\boldsymbol{\Phi}(\boldsymbol{x}_c) = [\varphi(s_1) \quad \varphi(s_2) \quad \ldots \quad \varphi(s_h)]^T \tag{6b}$$

$$\boldsymbol{s} = \boldsymbol{W}^h \boldsymbol{x}_c + \boldsymbol{b}^h \tag{6c}$$

where $\boldsymbol{W}^o \in \Re^{n \times h}$, $\boldsymbol{b}^o \in \Re^n$, $\boldsymbol{\Phi}(\boldsymbol{x}_c) \in \Re^h$ is the vector of hidden signals, $\boldsymbol{s} \in \Re^h$ is the vector of activation potentials, $\boldsymbol{W}^h \in \Re^{h \times n}$ is the matrix of hidden weighting factors, $\boldsymbol{b}^h \in \Re^h$ is the vector of hidden bias elements and h is the number of hidden neurons. In this work we use hyperbolic tangents as nonlinear activation functions, i.e., $\varphi(\cdot) = \tanh(\cdot)$. Vector $\boldsymbol{w}$ contains $\boldsymbol{b}^o$, $\boldsymbol{b}^h$, the columns of $\boldsymbol{W}^o$ and the columns of $\boldsymbol{W}^h$.

Our starting point for solving (4) using (6) is also a unit mapping, which is obtained by making $\boldsymbol{b}^o = \boldsymbol{0}$, $\boldsymbol{b}^h = \boldsymbol{0}$, $\boldsymbol{W}^h = 0.1[\boldsymbol{I}\ \boldsymbol{0}]^T$ and $\boldsymbol{W}^o = 10[\boldsymbol{I}\ \boldsymbol{0}]$, assuming that the training data has been scaled between -1 and $+1$. Notice that we consider $h \geq n$ in order to achieve the unit mapping.

The algorithm for finding the simplest inverse neuromapping is stated as follows

Algorithm: Inverse Neuromapping
begin solve (4) using (5) $h = n$ while $U_N(\boldsymbol{w}^*) > \varepsilon_L$ solve (4) using (6) $h = h+1$ end

In our implementation we use $\varepsilon_L = 1 \times 10^{-4}$. Notice that the algorithm for finding the inverse

neuromapping uses a 2-layer perceptron during at least the first n+1 NISM iterations, since the points $(\boldsymbol{x}_c^{(i)}, \boldsymbol{x}_f^{(i)})$ can be mapped with a linear mapping for $i = 1 \ldots n+1$. A 3-layer perceptron is needed only when we exceed n+1 NISM iterations and the mapping is significantly nonlinear.

E. Termination Criterion

As illustrated in the flow diagram of Fig. 1, we stop NISM optimization when the new iterate is close enough to the current point. We do this by testing the relative change in the fine model parameters. If the expression

$$\left\| \boldsymbol{x}_f^{(i+1)} - \boldsymbol{x}_f^{(i)} \right\|_2 \leq \varepsilon_{end} (1 + \varepsilon_{end} \left\| \boldsymbol{x}_f^{(i)} \right\|_2) \tag{7}$$

is true, we end NISM optimization taking $\boldsymbol{x}_f^{(i)}$ as the solution, otherwise we continue. We use $\varepsilon_{end} = 5\times10^{-3}$ in our implementation. Notice that the fine model is not evaluated at the point $\boldsymbol{x}_f^{(i+1)}$.

III. NATURE OF THE NISM STEP

In this section we prove that the NISM step, $\boldsymbol{x}_f^{(i+1)} = \boldsymbol{N}(\boldsymbol{x}_c^*)$, is equivalent to a quasi-Newton step while the inverse mapping built during NISM optimization remains linear, i.e., while a 2-layer perceptron is enough to approximate the inverse mapping. We also prove that the NISM step gradually departs from a quasi-Newton step as the amount of nonlinearity needed in the inverse mapping increases.

A. Jacobian of the Inverse Mapping

From (5), the Jacobian $\boldsymbol{J}_N$ of the inverse mapping $\boldsymbol{N}(\boldsymbol{x}_c)$ when a 2-layer perceptron is employed is given by

$$\boldsymbol{J}_N = \boldsymbol{W}^o \tag{8}$$

When a 3-layer perceptron is used, the Jacobian $\boldsymbol{J}_N$ is obtained from (6) as

$$\boldsymbol{J}_N = \boldsymbol{W}^o \boldsymbol{J}_{\boldsymbol{\Phi}} \boldsymbol{W}^h \tag{9}$$

where $\boldsymbol{J}_{\boldsymbol{\Phi}} \in \Re^{h\times h}$ is a diagonal matrix given by $\boldsymbol{J}_{\boldsymbol{\Phi}} = \text{diag}(\varphi'(s_j))$, with $j = 1 \ldots h$. We use (8) and (9) to demonstrate the nature of the NISM step $\boldsymbol{x}_f^{(i+1)} = \boldsymbol{N}(\boldsymbol{x}_c^*)$.

B. NISM Step vs. Quasi-Newton Step

A general space mapping optimization problem can be formulated as solving the system of

nonlinear equations

$$\boldsymbol{f}(\boldsymbol{x}_f) = \boldsymbol{P}(\boldsymbol{x}_f) - \boldsymbol{x}_c^* = \boldsymbol{0} \tag{10}$$

where $\boldsymbol{x}_c = \boldsymbol{P}(\boldsymbol{x}_f)$ is the mapping function that makes the coarse model behave as the fine model, i.e., $\boldsymbol{R}_c(\boldsymbol{P}(\boldsymbol{x}_f)) \approx \boldsymbol{R}_f(\boldsymbol{x}_f)$. A Newton step for solving (10) is given by

$$\boldsymbol{x}_f^{(i+1)} = \boldsymbol{x}_f^{(i)} - \boldsymbol{J}_P^{-1}\boldsymbol{f} \tag{11}$$

where $\boldsymbol{J}_P \in \Re^{n\times n}$ is the Jacobian of the mapping function $\boldsymbol{P}(\boldsymbol{x}_f)$. This can be stated in an equivalent manner by using the Jacobian $\boldsymbol{J}_N \in \Re^{n\times n}$ of the inverse of the mapping $\boldsymbol{x}_f = \boldsymbol{N}(\boldsymbol{x}_c)$ (see appendix A)

$$\boldsymbol{x}_f^{(i+1)} = \boldsymbol{x}_f^{(i)} - \boldsymbol{J}_N\boldsymbol{f} \tag{12}$$

Approximating $\boldsymbol{J}_N$ directly involves the same computational effort as approximating $\boldsymbol{J}_P$, but calculating the next step using (11) is computationally much more efficient than using (10), where a system of linear equations, possibly ill-conditioned, must be solved.

If a 2-layer perceptron is being used, we substitute (8) in (12) to obtain

$$\boldsymbol{x}_f^{(i+1)} = \boldsymbol{x}_f^{(i)} - \boldsymbol{W}^o(\boldsymbol{x}_c^{(i)} - \boldsymbol{x}_c^*) \tag{13}$$

which can be expressed using (5) as

$$\boldsymbol{x}_f^{(i+1)} = \boldsymbol{W}^o\boldsymbol{x}_c^* - (\boldsymbol{x}_f^{(i)} - \boldsymbol{b}^o) + \boldsymbol{x}_f^{(i)} = \boldsymbol{N}(\boldsymbol{x}_c^*) \tag{14}$$

From (12) and (14) we conclude that while the inverse mapping built during NISM optimization remains linear, the NISM step is equivalent to a quasi-Newton step. Notice that we do not use any of the classical updating formulae to calculate an approximation of the inverse of the Jacobian; this is done by simply evaluating the current neural network at the optimal coarse solution.

If a 3-layer perceptron is being used, we substitute (9) in (12) to obtain

$$\boldsymbol{x}_f^{(i+1)} = \boldsymbol{x}_f^{(i)} - \boldsymbol{W}^o\boldsymbol{J}_{\boldsymbol{\Phi}}\boldsymbol{W}^h(\boldsymbol{x}_c^{(i)} - \boldsymbol{x}_c^*) \tag{15}$$

Adding and subtracting $\boldsymbol{W}^o\boldsymbol{J}_{\boldsymbol{\Phi}}\boldsymbol{b}^h$ to (15)

$$\boldsymbol{x}_f^{(i+1)} = \boldsymbol{W}^o\boldsymbol{J}_{\boldsymbol{\Phi}}(\boldsymbol{W}^h\boldsymbol{x}_c^* + \boldsymbol{b}^h) - \boldsymbol{W}^o\boldsymbol{J}_{\boldsymbol{\Phi}}(\boldsymbol{W}^h\boldsymbol{x}_c^{(i)} + \boldsymbol{b}^h) + \boldsymbol{x}_f^{(i)} \tag{16}$$

Substituting (6c) in (16)

$$\boldsymbol{x}_f^{(i+1)} = \boldsymbol{W}^o \boldsymbol{J}_{\boldsymbol{\Phi}} \boldsymbol{s}(\boldsymbol{x}_c^*) - \boldsymbol{W}^o \boldsymbol{J}_{\boldsymbol{\Phi}} \boldsymbol{s}(\boldsymbol{x}_c^{(i)}) + \boldsymbol{x}_f^{(i)} \tag{17}$$

Expanding the term $\boldsymbol{J}_{\boldsymbol{\Phi}}\boldsymbol{s}(\boldsymbol{x}_c)$ we obtain

$$\boldsymbol{J}_{\boldsymbol{\Phi}} \boldsymbol{s}(\boldsymbol{x}_c) = [\varphi'(s_1)s_1 \quad \ldots \quad \varphi'(s_h)s_h]^T . \tag{18}$$

Since we are using hyperbolic tangents as nonlinear activation functions, when a small amount of nonlinearity is present (e.g., $s_j < 0.1$), $\varphi(s_j) = s_j$, and $\varphi'(s_j)s_j = s_j = \varphi(s_j)$, for $j = 1,\ldots, h$, and using (6b) we express (18) as

$$\boldsymbol{J}_{\boldsymbol{\Phi}} \boldsymbol{s}(\boldsymbol{x}_c) = \boldsymbol{\Phi}(\boldsymbol{x}_c) \tag{19}$$

Substituting (19) in (17)

$$\boldsymbol{x}_f^{(i+1)} = \boldsymbol{W}^o \boldsymbol{\Phi}(\boldsymbol{x}_c^*) - \boldsymbol{W}^o \boldsymbol{\Phi}(\boldsymbol{x}_c^{(i)}) + \boldsymbol{x}_f^{(i)} \tag{20}$$

Adding and subtracting $\boldsymbol{b}^o$ to (20) and using (6a) we express (20) as

$$\boldsymbol{x}_f^{(i+1)} = \boldsymbol{W}^o \boldsymbol{\Phi}(\boldsymbol{x}_c^*) + \boldsymbol{b}^o - \boldsymbol{W}^o \boldsymbol{\Phi}(\boldsymbol{x}_c^{(i)}) - \boldsymbol{b}^o + \boldsymbol{x}_f^{(i)} = \boldsymbol{N}(\boldsymbol{x}_c^*) \tag{21}$$

In conclusion, the NISM step gradually departs from a quasi-Newton step as the amount of nonlinearity needed in the inverse mapping increases.

IV. EXAMPLES

A. Two-Section Impedance Transformer

As an illustrative case, consider the classical test problem of designing a capacitively-loaded 10:1 two-section impedance transformer [17]. The proposed "coarse" and "fine" models are shown in Fig. 2. The "coarse" model consists of ideal transmission lines, while the "fine" model consists of capacitively-loaded ideal transmission lines, with $C_1 = C_2 = C_3 = 10$pF. The design specifications are $|S_{11}| \leq 0.50$ for frequencies between 0.5 GHz and 1.5 GHz.

The electrical lengths of the two transmission lines at 1.0 GHz are selected as design parameters. The characteristic impedances are kept fixed at the following values: $Z_1 = 2.23615\ \Omega$, $Z_2 = 4.47230\ \Omega$. Both models were implemented in OSA90/hope [18]. The optimal coarse solution is $\boldsymbol{x}_c^* = [90 \quad 90]^T$

(degrees). The coarse and fine model responses at $\boldsymbol{x}_c^*$ are shown in Fig. 3. We used only 10 frequency points from 0.2 to 1.8 GHz for the "fine" model.

NISM optimization requires only 3 "fine" model evaluations to solve this problem. The values of the fine model parameters at each iteration are shown in Table I. A 2-layer perceptron was enough to approximate the inverse mapping at all NISM iterations. The fine model response at the NISM solution is compared with the optimal coarse model response in Fig. 4. The fine model minimax objective function values at each NISM iteration are shown in Fig. 5.

Since both the coarse and "fine" models are actually very fast to evaluate, we applied direct minimax optimization to the "fine" model, obtaining $\boldsymbol{x}_f^* = [79.2651 \quad 74.2322]^T$ after 64 "fine" model evaluations. In Fig. 6 we compare the fine model response at this solution with the optimal NISM response, where a remarkable match is observed.

The same problem was solved in [19] using Trust Region Aggressive Space Mapping exploiting Surrogates. It is noticed that this algorithm required 7 "fine" model evaluations.

B. Bandstop Microstrip Filter with Open Stubs

We apply NISM optimization to a bandstop microstrip filter with quarter-wave resonant open stubs [11], whose physical structure is illustrated in Fig. 7. L_1, L_2 are the open stub lengths and W_1, W_2 the corresponding widths. An alumina substrate with thickness $H = 25$ mil, width $W_0 = 25$ mil and dielectric constant $\varepsilon_r = 9.4$ is used for a 50 Ω feeding line.

The specifications are $|S_{21}| \leq 0.01$ in the stopband and $|S_{21}| \geq 0.9$ in the passband, where the stopband lies between 9.3 GHz and 10.7 GHz, and the passband includes frequencies below 8 GHz and above 12 GHz. The design parameters are $\boldsymbol{x}_f = [W_1 \; W_2 \; L_0 \; L_1 \; L_2]^T$.

Sonnet's ***em***™ [20] driven by Empipe™ [18] was employed as the fine model, using a high-resolution grid with a 1mil×1mil cell size. The coarse model, illustrated in Fig. 8, consists of simple ideal transmission lines for modeling each microstrip section and classical formulas [21] to calculate the characteristic impedance and the effective dielectric constant of each transmission line. It is seen that L_{c2}

$= L_2 + W_0/2$, $L_{c1} = L_1 + W_0/2$, and $L_{c0} = L_0 + W_1/2 + W_2/2$. We use OSA90/hope™ [18] built-in transmission line elements TRL.

The following optimal coarse model solution is found for L_0, L_1, and L_2 of quarter-wave lengths at 10 GHz: $\boldsymbol{x}_c^* = [6.0 \;\; 9.0 \;\; 106.4 \;\; 110.1 \;\; 108.8]^T$ (mils), as in [11]. The coarse and fine model responses at the optimal coarse solution are shown in Fig. 9.

NISM optimization requires only 4 fine model evaluations to solve this problem. The sequence of iterates is shown in Table II (all the points are on the grid, to avoid interpolation). A 2-layer perceptron was enough to approximate the inverse mapping at all NISM iterations. The fine model response at the NISM solution is compared with the optimal coarse model response in Fig. 10. The fine model minimax objective function values at each NISM iteration are shown in Fig. 11.

The same problem was solved in [11] using Neural Space Mapping (NSM) optimization. NSM required 13 fine model evaluations to find the solution whose response is shown in Fig. 12. It is remarkable that NISM optimization not only requires fewer fine model evaluations, but also arrives at a solution closer to the solution of the original optimization problem.

C. High Temperature Superconducting Microstrip Filter

We apply NISM optimization to a high-temperature superconducting (HTS) quarter-wave parallel coupled-line microstrip filter, and contrast our results with those obtained by using NSM optimization on the same problem [10-11]. The physical structure of the HTS filter is illustrated in Fig. 13.

L_1, L_2 and L_3 are the lengths of the parallel coupled-line sections and S_1, S_2 and S_3 are the gaps between the sections. The width W is the same for all the sections as well as for the input and output lines, of length L_0. A lanthanum aluminate substrate with thickness H and dielectric constant ε_r is used.

The specifications are $|S_{21}| \geq 0.95$ in the passband and $|S_{21}| \leq 0.05$ in the stopband, where the stopband includes frequencies below 3.967 GHz and above 4.099 GHz, and the passband lies in the range [4.008GHz, 4.058GHz]. The design parameters are $\boldsymbol{x}_f = [L_1 \; L_2 \; L_3 \; S_1 \; S_2 \; S_3]^T$. We take $L_0 = 50$ mil, $H = 20$ mil, W = 7 mil, $\varepsilon_r = 23.425$, loss tangent = 3×10^{-5}; the metalization is considered lossless.

Sonnet's ***em***™ [20] driven by Empipe™ [18] was employed as the fine model, using a high-resolution grid with a 1mil×1mil cell size. OSA90/hope™ [18] built-in linear elements MSL (microstrip line), MSCL (two-conductor symmetrical coupled microstrip lines) and OPEN (open circuit) connected by circuit theory over the same MSUB (microstrip substrate definition) are taken as the "coarse" model, whose schematic representation is illustrated in Fig. 14.

The following optimal coarse model solution is used, as in [10,11]: $\boldsymbol{x}_c^* = [188.33 \;\; 197.98 \;\; 188.58 \;\; 21.97 \;\; 99.12 \;\; 111.67]^T$ (in mils). The coarse and fine model responses at the optimal coarse solution are shown in Fig. 15. Only 14 frequency points per frequency sweep are used for the fine model.

After only 3 fine model simulations the optimal NISM solution was found. The sequence of fine model parameters at each NISM iteration is shown in Table III (all the points are on the grid, to avoid interpolation). A 2-layer perceptron was enough to approximate the inverse mapping at all NISM iterations. Fig. 16a compares the optimal coarse response with the fine model response at the NISM solution using a fine frequency sweep. A more detailed comparison in the passband is shown in Fig. 16b. The fine model minimax objective function values at each NISM iterations for this problem are shown in Fig. 17.

The same problem was solved in [19] using Trust Region Aggressive Space Mapping exploiting Surrogates. It is noticed that this algorithm required 8 fine model evaluations; the corresponding fine model minimax objective function values are shown in Fig. 18.

Fig. 19 shows the results obtained by applying NSM optimization [11] to the same problem, where the optimal NSM solution was found after 14 fine model evaluations.

Once again, it is seen that NISM optimization is not only more efficient in terms of the required fine model evaluations, but also yields a solution closer to the optimal solution of the original optimization problem (compare Fig. 16b with Fig. 19, and Fig. 17 with Fig. 18).

For all the previous examples, parameter extraction was successfully performed in just one attempt at every NISM iteration. That was not the case for the HTS filter, where the parameter extraction

objective function has many poor local minima around $\boldsymbol{x}_c^*$. Our proposed algorithm for parameter extraction overcame this problem. We applied NISM optimization to the HTS filter 5 times in order to test the statistical parameter extraction results. In Table IV we show the number of attempts needed for successful parameter extraction at each NISM iteration for the 5 optimizations. Exactly the same sequence of points illustrated in Table III was predicted by each of the 5 optimizations.

V. CONCLUSIONS

We propose Neural Inverse Space Mapping (NISM) optimization for EM-based design of microwave structures. The inverse of the mapping is exploited for the first time in a space mapping algorithm. NISM optimization does not require up-front EM simulations, multipoint parameter extraction or frequency mapping. A simple statistical procedure overcomes the existence of poor local minima during parameter extraction. A neural network whose generalization performance is controlled through a network growing strategy approximates the inverse of the mapping at each iteration. The NISM step simply evaluates the current neural network at the optimal coarse solution. We prove that this step is equivalent to a quasi-Newton step while the inverse mapping remains essentially linear, and gradually departs from a Newton step as the amount of nonlinearity in the inverse mapping increases. Our new algorithm exhibits superior performance to Neural Space Mapping (NSM) optimization and Trust Region Aggressive Space Mapping exploiting Surrogates.

ACKNOWLEDGEMENT

The authors thank Dr. J.C. Rautio, President, Sonnet Software, Inc., Liverpool, NY, for making ***em***™ available.

REFERENCES

[1] P. Burrascano and M. Mongiardo, "A review of artificial neural networks applications in microwave CAD," *Int. J. RF and Microwave CAE*, Special Issue on Applications of ANN to RF and Microwave Design, vol. 9, 1999, pp. 158-174.

[2] 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.

[3] T.S. Horng, C.C. Wang and N.G. Alexopoulos, "Microstrip circuit design using neural networks," *IEEE MTT-S Int. Microwave Symp. Dig.* (Atlanta, GA), 1993, pp. 413-416.

[4] A.H. Zaabab, Q.J. Zhang and M.S. Nakhla, "A neural network modeling approach to circuit optimization and statistical design," *IEEE Trans. Microwave Theory Tech.*, vol. 43, 1995, pp. 1349-1358.

[5] A. Veluswami, M.S. Nakhla and Q.J. Zhang, "The application of neural networks to EM-based simulation and optimization of interconnects in high-speed VLSI circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 45, 1997, pp. 712-723.

[6] P.M. Watson and K.C. Gupta, "Design and optimization of CPW circuits using EM-ANN models for CPW components," *IEEE Trans. Microwave Theory Tech.*, vol. 45, 1997, pp. 2515-2523.

[7] P. Burrascano, M. Dionigi, C. Fancelli and M. Mongiardo, "A neural network model for CAD and optimization of microwave filters," *IEEE MTT-S Int. Microwave Symp. Dig.* (Baltimore, MD), 1998, pp. 13-16.

[8] P.M. Watson, G.L. Creech and K.C. Gupta, "Knowledge based EM-ANN models for the design of wide bandwidth CPW patch/slot antennas," *IEEE AP-S Int. Symp. Digest* (Orlando, FL), 1999, pp. 2588-2591.

[9] C. Cho and K.C. Gupta, "EM-ANN modeling of overlapping open-ends in multilayer lines for design of bandpass filters," *IEEE AP-S Int. Symp. Digest* (Orlando, FL), 1999, pp. 2592-2595.

[10] M.H. Bakr, J.W. Bandler, M.A. Ismail, J.E. Rayas-Sánchez and Q.J. Zhang, "Neural space mapping optimization of EM microwave structures," *IEEE MTT-S Int. Microwave Symp*. *Digest* (Boston, MA), 2000, pp. 879-882.

[11] M.H. Bakr, J.W. Bandler, M.A. Ismail, J.E. Rayas-Sánchez and Q.J. Zhang, "Neural space mapping optimization for EM-based design," *IEEE Trans. Microwave Theory Tech*., vol. 48, 2000, pp. 2307-2315.

[12] J.W. Bandler, M.A. Ismail, J.E. Rayas-Sánchez and Q.J. Zhang, "Neural inverse space mapping EM-optimization," *IEEE MTT-S Int. Microwave Symp*. *Digest* (Phoenix, AZ), 2001.

[13] Matlab™ Optimization Toolbox, Version 2, The MathWorks, Inc., 3 Apple Hill Drive, Natick MA 01760-2098, 1999.

[14] J.W. Bandler, R.M. Biernacki, S.H. Chen and D. Omeragić, "Space mapping optimization of waveguide filters using finite element and mode-matching electromagnetic simulators," *Int. J. RF and Microwave CAE*, vol. 9, 1999, pp. 54-70.

[15] Matlab™ Neural Network Toolbox, Version 3, The MathWorks, Inc., 3 Apple Hill Drive, Natick MA 01760-2098, 1998.

[16] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New Jersey, MA: Prentice Hall, 1999.

[17] J.W. Bandler, "Optimization methods for computer-aided design," *IEEE Trans. Microwave Theory Tech*., vol. 17, 1969, pp. 533-552.

[18] OSA90/hope™ and Empipe™, Version 4.0, formerly Optimization Systems Associates Inc., P.O. Box 8083, Dundas, ON, Canada, L9H 5E7, 1997, now Agilent Technologies, 1400 Fountaingrove Parkway, Santa Rosa, CA 95403-1799.

[19] M.H. Bakr, J.W. Bandler, K. Madsen, J.E. Rayas-Sánchez and J. Søndergaard, "Space mapping optimization of microwave circuits exploiting surrogate models," *IEEE Trans. Microwave Theory Tech*., vol. 48, 2000, pp. 2297-2306.

[20] ***em***™ Version 4.0b*,* Sonnet Software, Inc., 1020 Seventh North Street, Suite 210, Liverpool, NY 13088, 1997.

[21] M. Pozar, *Microwave Engineering*. Amherst, MA: John Wiley and Sons, 1998, pp. 162.

Appendix A

$\boldsymbol{J}_N = \boldsymbol{J}_P^{-1}$, Proof

Let $\boldsymbol{x}_c = \boldsymbol{P}(\boldsymbol{x}_f)$, with $\boldsymbol{P}$: $\Re^n \rightarrow \Re^n$, and $\boldsymbol{x}_f = \boldsymbol{N}(\boldsymbol{x}_c)$ its inverse function. Using $\boldsymbol{P}(\boldsymbol{x}_f)$, we can write the system of equations

$$\begin{aligned} dx_{c1} &= \frac{\partial x_{c1}}{\partial x_{f1}} dx_{f1} + \frac{\partial x_{c1}}{\partial x_{f2}} dx_{f2} + \ldots + \frac{\partial x_{c1}}{\partial x_{fn}} dx_{fn} \\ dx_{c2} &= \frac{\partial x_{c2}}{\partial x_{f1}} dx_{f1} + \frac{\partial x_{c2}}{\partial x_{f2}} dx_{f2} + \ldots + \frac{\partial x_{c2}}{\partial x_{fn}} dx_{fn} \\ &\vdots \\ dx_{cn} &= \frac{\partial x_{cn}}{\partial x_{f1}} dx_{f1} + \frac{\partial x_{cn}}{\partial x_{f2}} dx_{f2} + \ldots + \frac{\partial x_{cn}}{\partial x_{fn}} dx_{fn} \end{aligned} \tag{A-1}$$

Let us define

$$\boldsymbol{dx}_c = [dx_{c1} \quad \ldots \quad dx_{cn}]^T \tag{A-2}$$

$$\boldsymbol{dx}_f = [dx_{f1} \quad \ldots \quad dx_{fn}]^T \tag{A-3}$$

$$\boldsymbol{J}_P = \begin{bmatrix} \frac{\partial x_{c1}}{\partial x_{f1}} & \frac{\partial x_{c1}}{\partial x_{f2}} & \cdots & \frac{\partial x_{c1}}{\partial x_{fn}} \\ \frac{\partial x_{c2}}{\partial x_{f1}} & \frac{\partial x_{c2}}{\partial x_{f2}} & \cdots & \frac{\partial x_{c2}}{\partial x_{fn}} \\ \vdots & & & \\ \frac{\partial x_{cn}}{\partial x_{f1}} & \frac{\partial x_{cn}}{\partial x_{f2}} & \cdots & \frac{\partial x_{cn}}{\partial x_{fn}} \end{bmatrix} \tag{A-4}$$

Substituting (A-2)-(A-4) in (A-1)

$$\boldsymbol{dx}_c = \boldsymbol{J}_P \, \boldsymbol{dx}_f \tag{A-5}$$

Similarly, using $\boldsymbol{N}(\boldsymbol{x}_c)$ we obtain

$$\boldsymbol{dx}_f = \boldsymbol{J}_N \, \boldsymbol{dx}_c \tag{A-6}$$

where

$$\boldsymbol{J}_N = \begin{bmatrix} \dfrac{\partial x_{f1}}{\partial x_{c1}} & \dfrac{\partial x_{f1}}{\partial x_{c2}} & \cdots & \dfrac{\partial x_{f1}}{\partial x_{cn}} \\ \dfrac{\partial x_{f2}}{\partial x_{c1}} & \dfrac{\partial x_{f2}}{\partial x_{c2}} & \cdots & \dfrac{\partial x_{f2}}{\partial x_{cn}} \\ \vdots & & & \\ \dfrac{\partial x_{fn}}{\partial x_{c1}} & \dfrac{\partial x_{fn}}{\partial x_{c2}} & \cdots & \dfrac{\partial x_{fn}}{\partial x_{cn}} \end{bmatrix} \tag{A-7}$$

Comparing (A-5) and (A-6) we conclude that $\boldsymbol{J}_N = \boldsymbol{J}_P^{-1}$. Notice that when $\boldsymbol{x}_f$ and $\boldsymbol{x}_c$ have different dimensionality, $\boldsymbol{J}_N$ is the pseudoinverse of $\boldsymbol{J}_P$.

TABLE I
FINE MODEL PARAMETERS FOR THE
TWO-SECTION IMPEDANCE TRANSFORMER
AT EACH NISM ITERATION

i	$\boldsymbol{x}_f^{(i)\,T}$
1	[90 90]
2	[84.1990 83.0317]
3	[79.3993 73.7446]

TABLE II
FINE MODEL PARAMETERS FOR THE
BANDSTOP FILTER WITH OPEN STUBS
AT EACH NISM ITERATION

i	$\boldsymbol{x}_f^{(i)\,T}$
1	[6 9 106 110 109]
2	[7 11 103 112 111]
3	[9 20 95 115 115]
4	[9 19 95 115 114]

TABLE III
FINE MODEL PARAMETERS FOR THE
HTS MICROSTRIP FILTER
AT EACH NISM ITERATION

i	$\boldsymbol{x}_f^{(i)\,T}$
1	[188 198 189 22 99 112]
2	[187 196 187 21 84 92]
3	[186 194 185 20 80 89]

TABLE IV
PARAMETER EXTRACTION RESULTS FOR 5
NISM OPTIMIZATIONS FOR THE HTS FILTER

i	number of attempts needed for successfull PE				
1	12	9	3	10	8
2	3	3	6	7	3
3	1	1	1	1	1

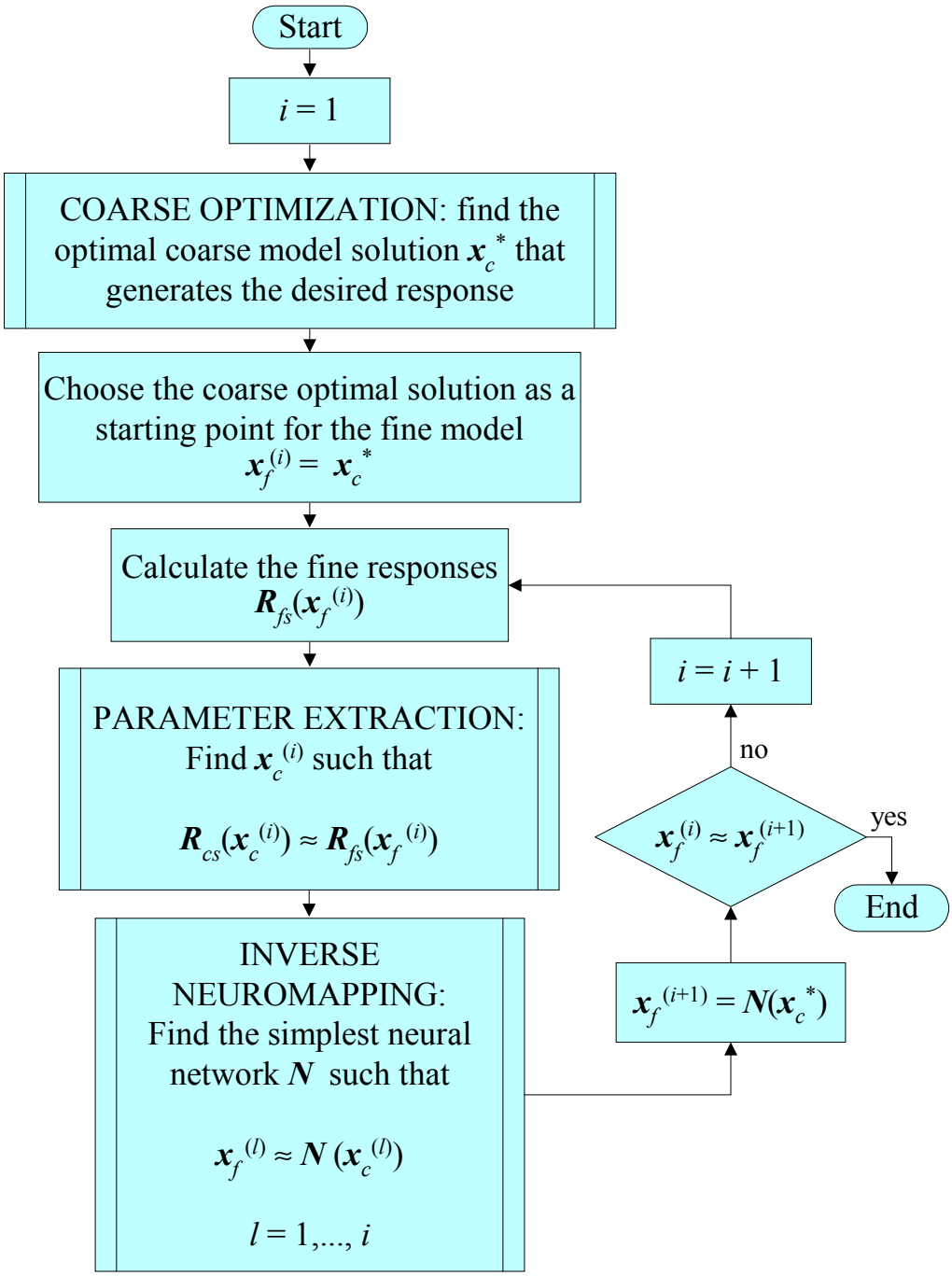

Fig. 1. Flow diagram for Neural Inverse Space Mapping (NISM) optimization.

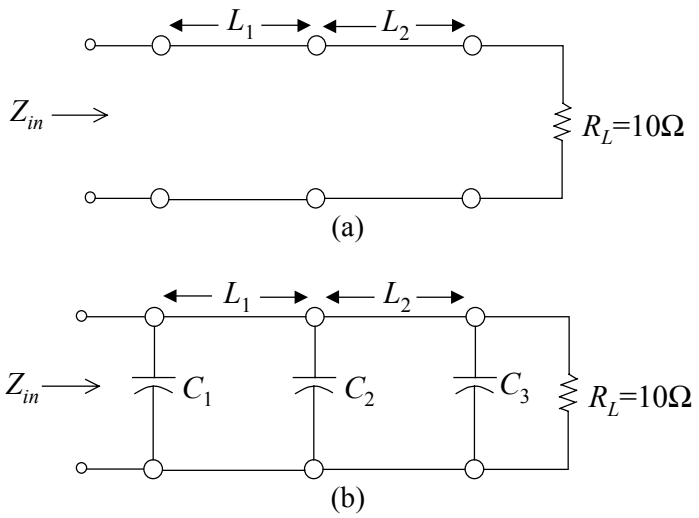

Fig. 2. Two-section impedance transformer test problem: (a) "coarse" model, (b) "fine" model.

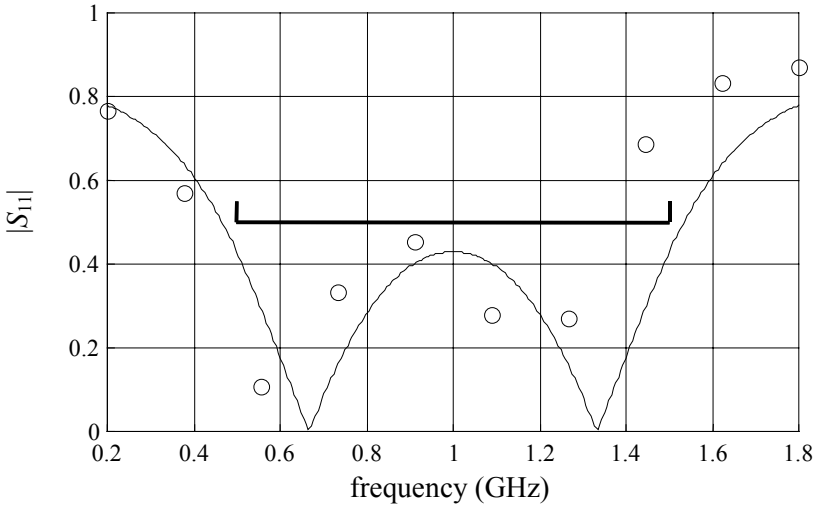

Fig. 3. Coarse (−) and fine (○) model responses at $\boldsymbol{x}_c^*$ for the two-section impedance transformer.

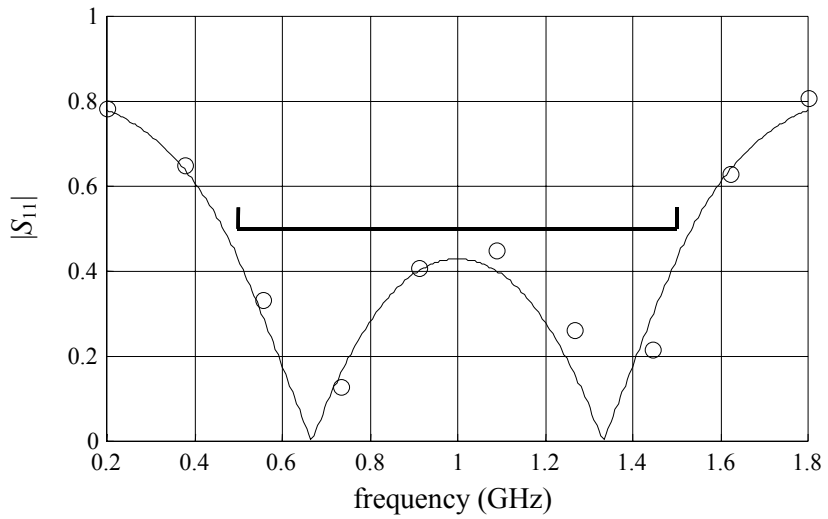

Fig. 4. Optimal coarse model response (–) and fine model response at the NISM solution (○) for the two-section impedance transformer.

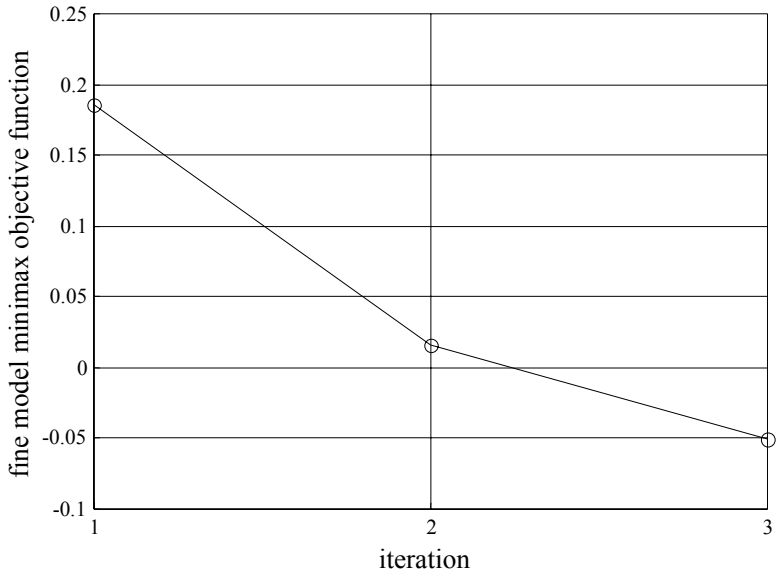

Fig. 5. Fine model minimax objective function values for the two-section impedance transformer at each NISM iteration.

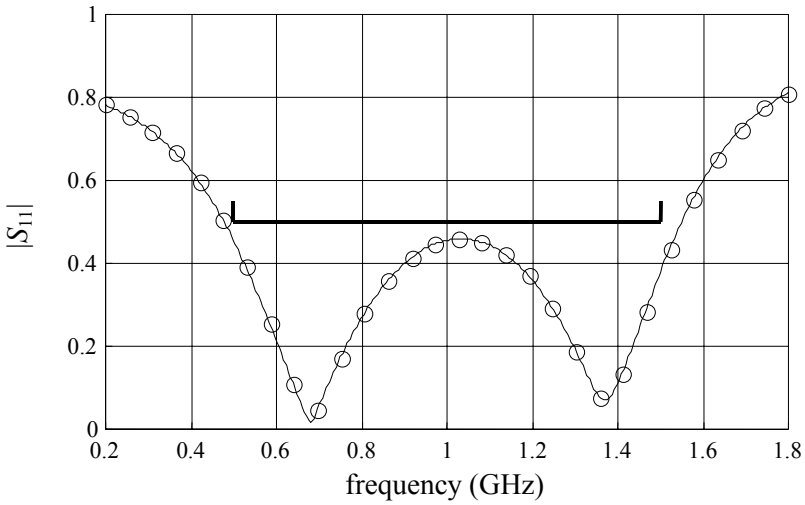

Fig. 6. Fine model response at the NISM solution (○) and at the direct minimax solution (–) for the two-section impedance transformer.

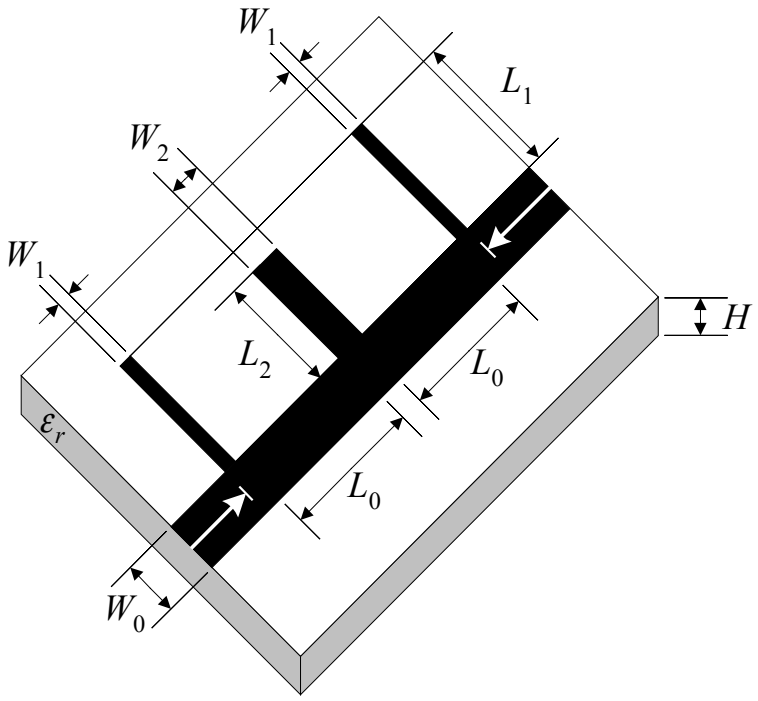

Fig. 7. Bandstop microstrip filter with quarter-wave resonant open stubs.

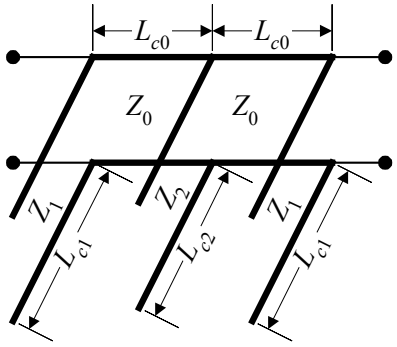

Fig. 8. Coarse model for the bandstop microstrip filter with open stubs.

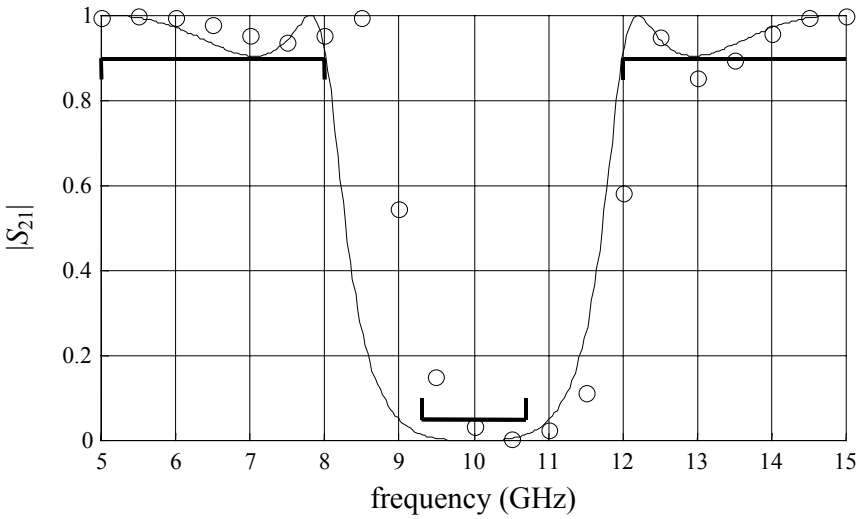

Fig. 9. Coarse and fine model responses at the optimal coarse solution for the bandstop filter with open stubs: OSA90/hope™ (–) and ***em***™ (○).

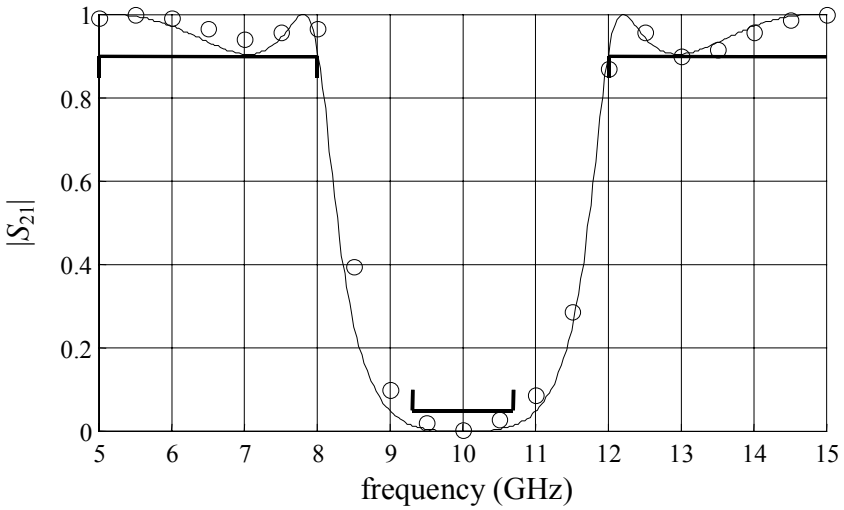

Fig. 10. Coarse model response (–) at the optimal coarse solution and fine model response (○) at the NISM solution for the bandstop microstrip filter with open stubs.

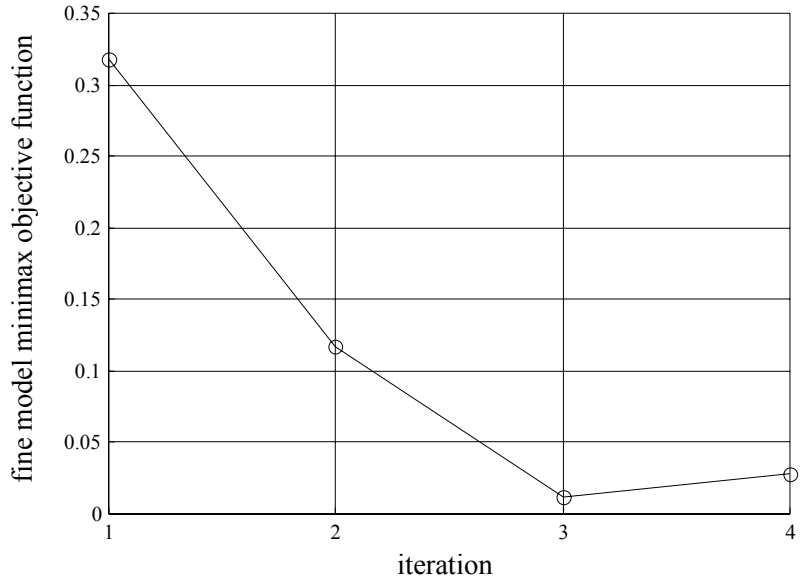

Fig. 11. Fine model minimax objective function values for the bandstop microstrip filter at each NISM iteration.

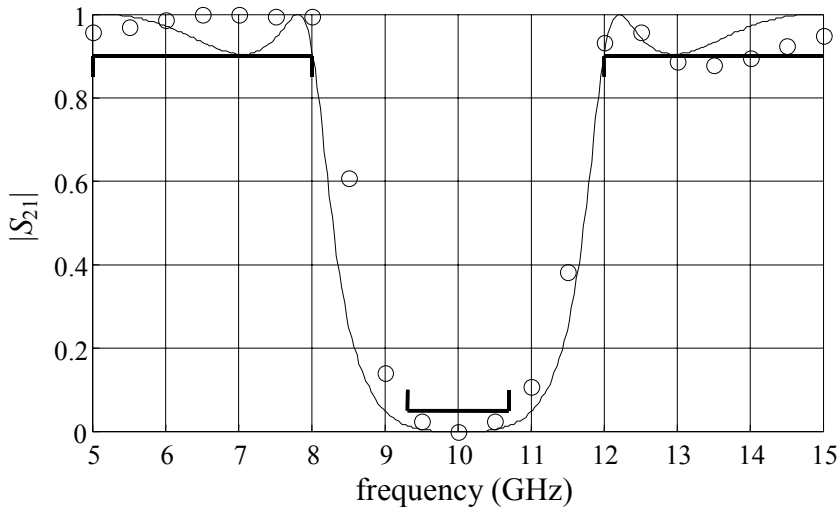

Fig. 12. Coarse model response (−) at the optimal coarse solution and fine model response (○) at the NSM solution, obtained in [2], for the bandstop microstrip filter with open stubs.

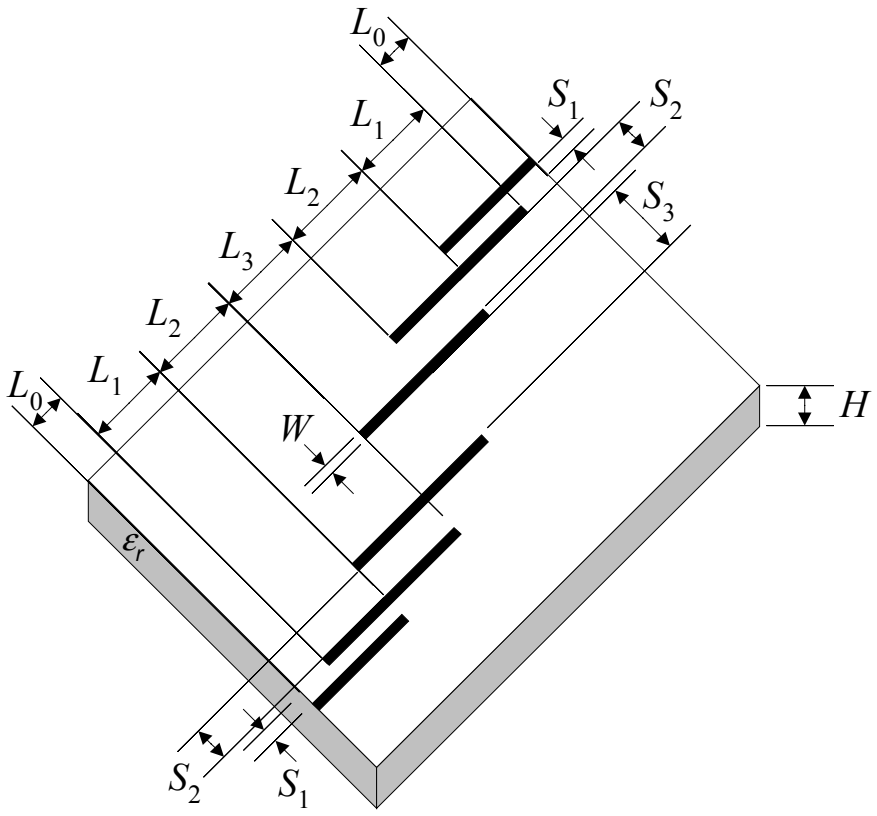

Fig. 13. HTS quarter-wave parallel coupled-line microstrip filter.

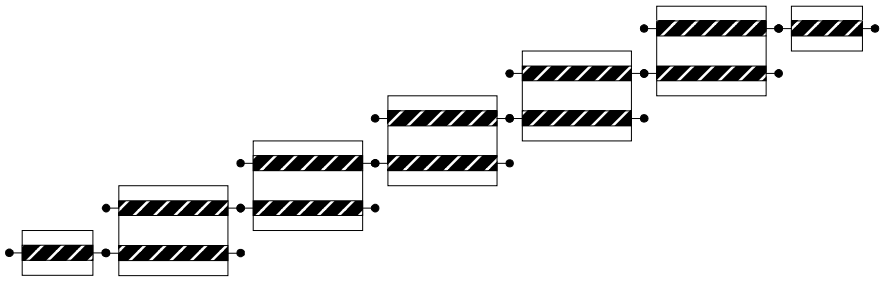

Fig. 14. Schematic representation of the coarse model for the HTS filter.

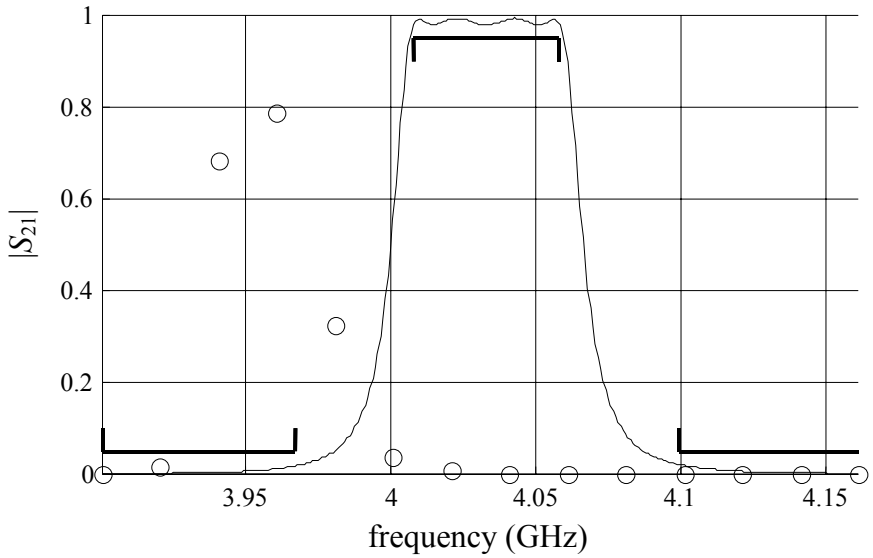

Fig. 15. Coarse and fine model responses at the optimal coarse solution for the HTS filter: OSA90/hope™ (–) and ***em***™ (○).

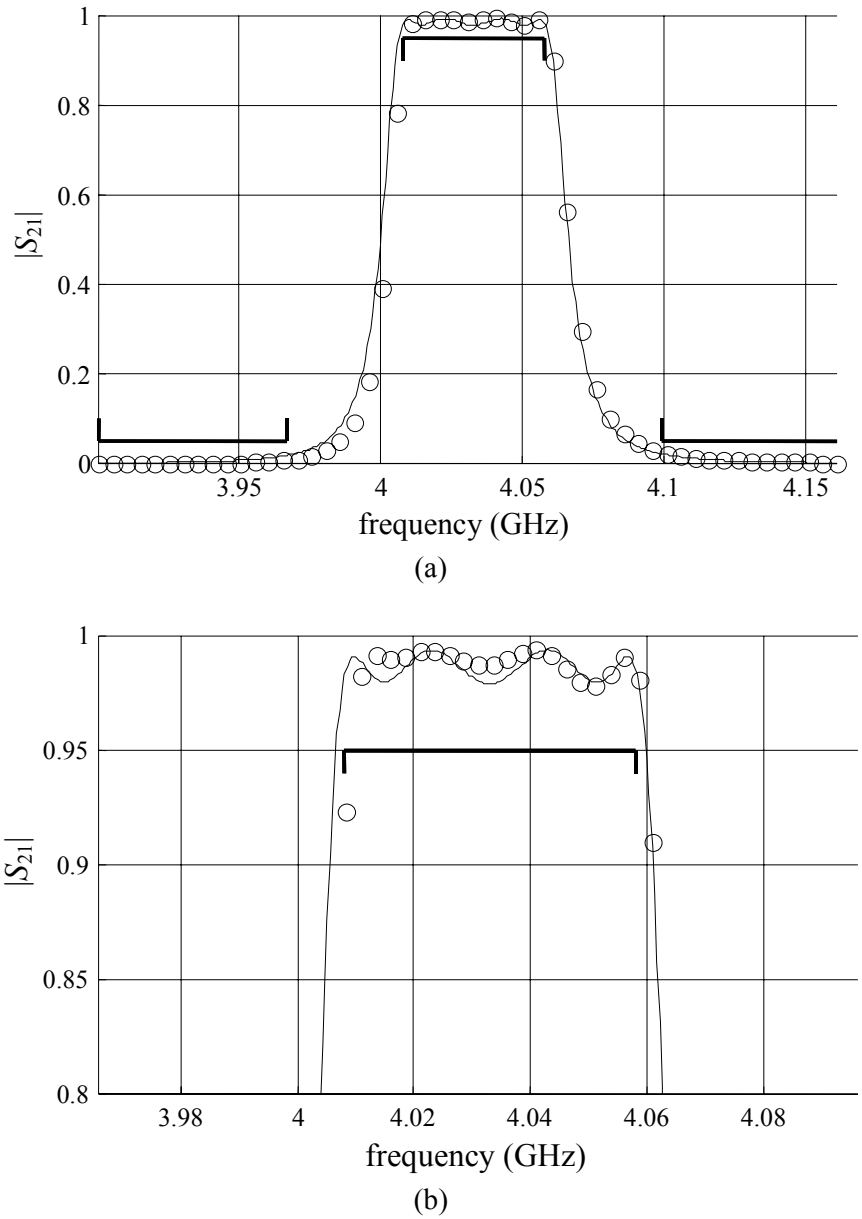

Fig. 16. Coarse model response at $\boldsymbol{x}_c^*$ (–) and fine model response at $\boldsymbol{x}_f^{NISM}$ (○) for the HTS filter: (a) in the complete range of interest, (b) in the passband.

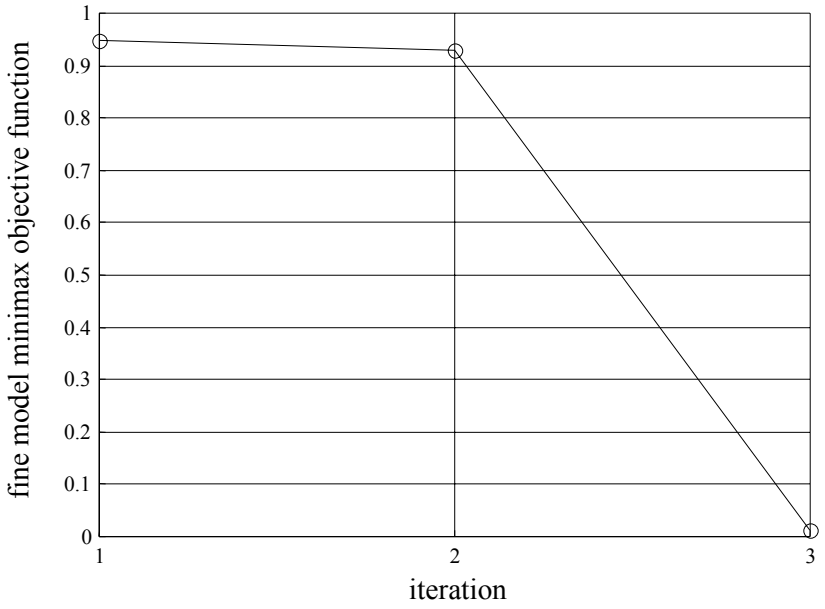

Fig. 17. Fine model minimax objective function values for the HTS microstrip filter at each NISM iteration.

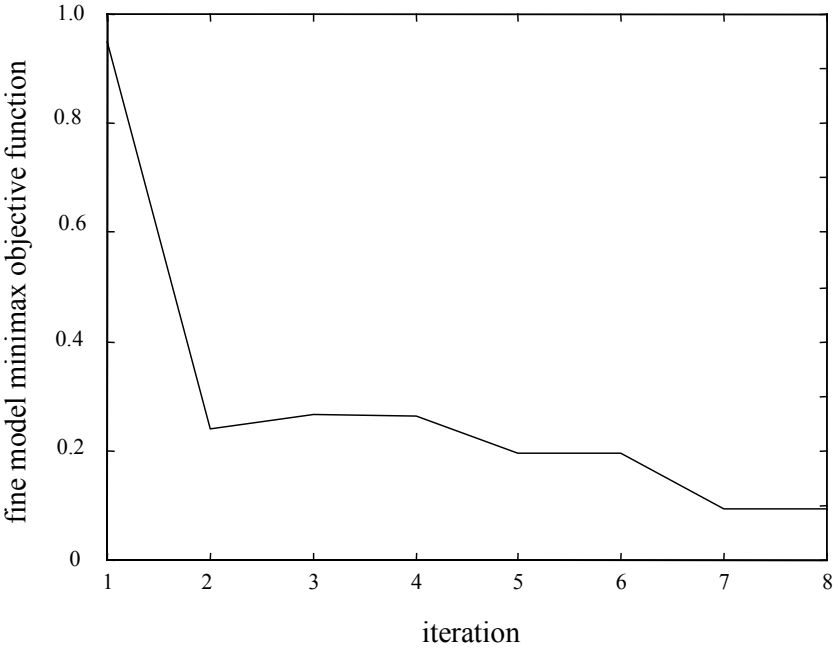

Fig. 18. Fine model minimax objective function values for the HTS microstrip filter at each iteration using Trust Region Aggressive Space Mapping exploiting Surrogates, as obtained in [19].

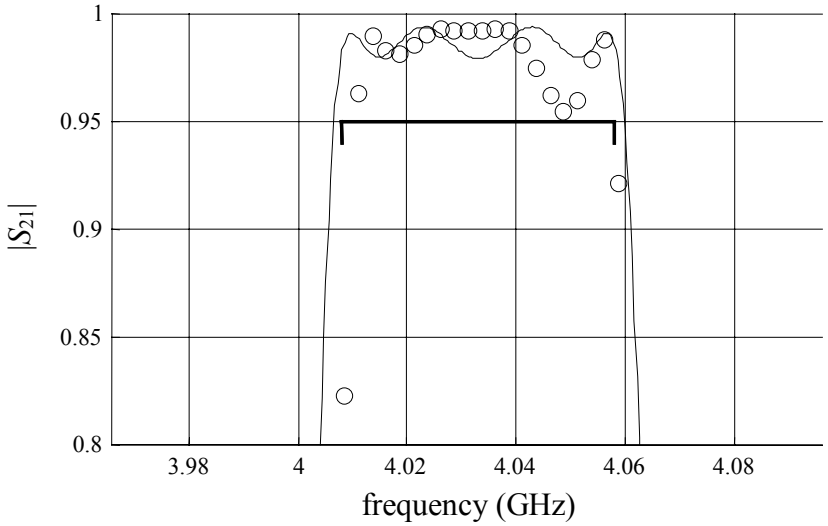

Fig. 19. Coarse model response at $\boldsymbol{x}_c^*$ (–) and fine mode response at $\boldsymbol{x}_f^{NSM}$ (○), obtained in [2], in the passband.