A Unified Automated Parametric Modeling Algorithm Using Knowledge-Based Neural Network and l_1 Optimization

Weicong Na, *Student Member, IEEE*, Feng Feng, *Student Member, IEEE*, Chao Zhang, *Student Member, IEEE*, and Qi-Jun Zhang, *Fellow, IEEE*

***Abstract*— Knowledge-based neural network modeling techniques using space-mapping concept have been demonstrated in the existing literature as efficient methods to overcome the accuracy limitations of empirical/equivalent circuit models when matching new electromagnetic data. For different modeling problems, the mapping structures can be different. In this paper, we propose a unified automated model generation algorithm that uses l_1 optimization to automatically determine the type and the topology of the mapping structure in a knowledge-based neural network model. By encompassing various types of mappings of the knowledge-based neural network model in the existing literature, we present a new unified model structure and derive new sensitivity formulas for the training of the unified model. The proposed l_1 formulation of modeling can force some weights of the mapping neural networks to zeros while leaving other weights as nonzeros. We utilize this feature to allow l_1 optimization to automatically determine which mapping is necessary and which mapping is unnecessary. Using the proposed l_1 optimization method, the mapping structure can be determined to address different needs of different modeling problems. The structure of the final knowledge-based model can be flexible combinations of some or all of linear mapping, nonlinear mapping, input mapping, frequency mapping, and output mapping. In this way, the proposed algorithm is more systematic and can further speed up the knowledge-based modeling process than existing knowledge-based modeling algorithms. The proposed method is illustrated by three microwave filter modeling examples.**

***Index Terms*— Design automation, electromagnetic (EM) modeling, knowledge-based neural network, microwave filter, parametric model.**

I. Introduction

ARTIFICIAL neural network (ANN) techniques have been recognized as powerful tools in microwave modeling and design [1]–[4], such as on-chip inductor modeling [5], power amplifier modeling [6]–[8], parametric microwave components modeling [9], [10], and electromagnetic (EM) optimization [11]. ANNs learn EM data through the training process and the trained neural networks are then used as fast and accurate models for efficient high-level circuit and system design. In the microwave modeling area, knowledge-based neural network modeling approaches using the space-mapping (SM) concept [1], [12], [13] have been described in the literature for obtaining a better model with limited data. The idea of the knowledge-based model is to exploit existing knowledge in the form of empirical or equivalent circuit models together with neural networks to develop a faster and more accurate model. For the microwave design, there exist many empirical or equivalent circuit models that are computationally efficient and widely used in practical design. However, such models are often valid only in a limited parameter range, beyond which the model predictions become inaccurate. As with the SM concept, the empirical/equivalent circuit models are considered as "coarse model" [14], [15] and the EM data is considered as "fine data." The SM technique is developed to use linear mappings to establish a mathematical link between the coarse model and the fine data [14], [15], and to obtain a faster and more accurate model. However, when the modeling range becomes large, linear mappings only are not enough. The neural networks are used to provide a nonlinear computational approach to bridge the gap between the empirical/equivalent circuit model and the new EM simulation data [13]. This is achieved with the SM concept using neural networks to represent the nonlinear mappings between the empirical/equivalent circuit model and the EM data. Extrapolation capability is also enhanced because of the embedded knowledge in the model [12]. For simplicity and convenience of the ongoing descriptions, we will use the term "empirical model" to imply the empirical and/or equivalent circuit models in the subsequent part of this paper.

By taking advantage of the vast set of empirical models already available, SM-based neural network models decrease the number of EM simulations for training, improve model accuracy and generalization ability, and reduce the complexity of the ANN topology with respect to the classical neural network modeling approach [13], [14]. A number of papers cover different kinds of SMs, including input SM [10], [14], frequency SM [16], and output SM [17]. Input SM is used to modify the values of the geometrical design variables to a

Manuscript received March 28, 2016; revised August 3, 2016 and October 3, 2016; accepted October 30, 2016. Date of publication December 8, 2016; date of current version March 3, 2017. This work was supported by the Natural Sciences and Engineering Research Council of Canada under Grant RGPIN 122049-09 and Grant STPGP447367-13.

W. Na and F. Feng are with the School of Electronic Information Engineering, Tianjin University, Tianjin 300072, China, and also with the Department of Electronics, Carleton University, Ottawa, ON K1S5B6, Canada (e-mail: weicongna@doe.carleton.ca; fengfeng@doe.carleton.ca).

C. Zhang and Q. J. Zhang are with the Department of Electronics, Carleton University, Ottawa, ON K1S5B6, Canada (e-mail: chaozhang@doe.carleton.ca; qjz@doe.carleton.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMTT.2016.2630059

different set of values to be supplied to the empirical model, so that the modified empirical model response can match the EM simulation data [14], [18]. For those cases where the shapes of the EM simulation and the empirical model response are nearly identical, but shifted in frequency, the frequency SM is used to align both the responses [16]. Output SM is used to enhance the robustness of the modeling process in case other SMs cannot provide sufficient matching with the EM data [17], [19]. Any of these three mappings can be either linear or nonlinear. For simple modeling problems within a small geometrical parameter range, linear mappings [15], [20] are used to obtain a good match between the model and the training data. For complicated modeling problems within a large geometrical parameter range, nonlinear mappings [21] are necessary in order to obtain an accurate model.

Several approaches for the structure selection of the knowledge-based model are described in the existing literature [20], [22]–[24]. In [20], both input and output SMs are used in the knowledge-based model for microwave device optimization. However, all the mappings in [20] are linear mappings, and the decision about whether to use input mapping or output mapping is made manually with the designer's knowledge of the problem and engineering experience. The method in [22] uses genetic algorithm to find suitable combinations of the mapping structure and the empirical model, but it cannot distinguish between linear mapping and nonlinear mapping.

In [25], automated model generation method is described to automate the structure selection process of the pure neural network modeling. In [23], automated model generation method is expanded from generating pure neural network models to generating knowledge-based models. It integrates all the subtasks in generating a knowledge-based model into one framework and further reduces the number of training data required. However, the structure of the knowledge-based model in [23] is fixed at a predetermined structure and the automated algorithm is more focused on the automation of data sampling process. The model structure in [23] is a combination of an empirical model and a nonlinear input mapping which is realized by a three-layer multilayer perceptron (MLP) neural network. However, the selection of different mapping structures is not addressed in [23]. For some complicated EM modeling problems, input mapping only may not be good enough to achieve an accurate model. On the other hand, for some simple modeling problems, a nonlinear input mapping is redundant and only one simple linear mapping may be good enough to meet the accuracy requirement.

Different mapping structures should be needed for different modeling problems. The mapping structure depends on many factors, such as the complexity of the specific modeling problem, the quality of the empirical model, and the modeling range of geometrical parameters. For example, for a given set of EM data, different empirical models may need different mapping structures. For another example, if an empirical model is to be mapped to different sets of EM data with different ranges of geometrical parameters, the mapping structures in the final model must be different. Since the mapping structure is problem dependent, the development of the automated model structure adaptation algorithm is very important.

Most recently, preliminary work for a relatively flexible automated model structure adaptation method for knowledge-based model development is described in [24]. It takes both input and output mapping, and linear and nonlinear mapping into consideration during the modeling process. The final knowledge-based model can be any combination of the empirical model and the mapping neural networks. However, when determining the mapping structure, the algorithm is based on a brute force sequential trial and error mechanism, first trying input mapping and then output mapping, first trying linear mapping and then nonlinear mapping. It compares various combinations of mappings and finds a suitable knowledge-based neural network model. The process involves many trial and error computation steps and is usually time-consuming.

In this paper, we propose a more elegant and unified automated model structure adaptation algorithm for knowledge-based parametric modeling. The proposed technique is a substantial advance over [24]. We propose a new formulation using l_1 optimization to automatically determine all the mappings in the knowledge-based model. We propose an extended and unified model structure to encompass various types of mappings. New sensitivity formulas for the training of the unified model are also proposed and derived in this paper. The use of the l_1 optimization, based on its theoretical properties [26], [27], is a critical part of the proposed training algorithm and optimization process. The l_1 optimization has the distinctive property for feature selection within the training process. At the end of l_1 optimization, some weights in the mapping neural networks are zeros while others remain nonzeros. Zero weights mean that the corresponding parts of the mapping can be ignored and deleted. Using this property, we formulate l_1 optimization solutions to indicate whether a mapping is linear or nonlinear, and whether a mapping should be input mapping, frequency mapping, or output mapping. Compared to traditional knowledge-based models with a fixed mapping structure, the proposed method can automatically adjust the mapping structure to achieve an accurate model with the most suitable and compact structure. Compared to the existing literature on the model structure selection method, our proposed method is a more systematic technique and can further speed up the process of developing a knowledge-based neural network model.

In Section II, we propose a unified and general knowledge-based neural network structure. In Section III, we describe the proposed training method with l_1 optimization and the computation of error derivatives needed during the training process. A summary and flowchart of the proposed algorithm are presented in Section III. The proposed method is illustrated by three microwave filter modeling examples in Section IV.

II. Proposed Unified Knowledge-Based Model Structure

Let $\boldsymbol{x} = [x_1, x_2, \ldots, x_N]^T$ represent a vector of the inputs of the proposed knowledge-based model. We define N as the number of the model inputs. $\boldsymbol{x}$ contains the physical geometrical parameters of a microwave device, such as the length

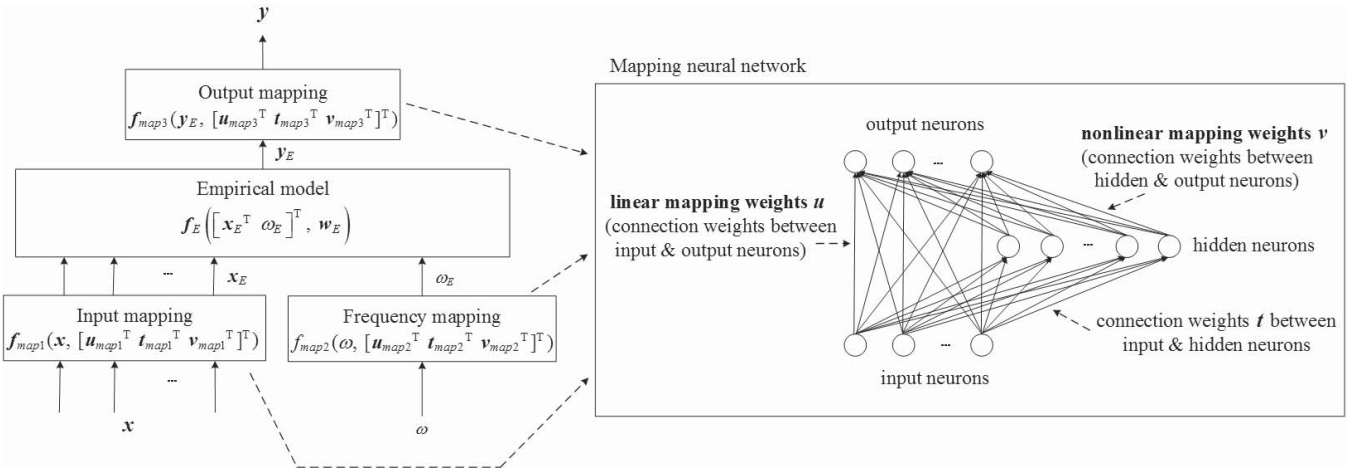

Fig. 1. Proposed unified knowledge-based model structure. It combines the empirical model with three mapping neural networks. Each mapping neural network contains linear mapping and nonlinear mapping.

and the width of an EM structure. Let $\boldsymbol{y} = [y_1, y_2, \ldots, y_M]^T$ represent a vector of the outputs of the knowledge-based neural network model, such as S-parameters. We define M as the number of the model outputs. Empirical models often exist to represent the relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$. However, the accuracy of the empirical model is usually limited, especially when the values of the EM geometrical parameters change. The mapping method using neural networks is utilized to address the situation when an existing empirical model cannot fit the new EM data well. The mapping structures are affected by many factors and the determination of the mapping structures is not straightforward. In [24], the knowledge-based model structure combines the empirical model with one input mapping, one frequency mapping, and one output mapping. The three mappings can be either a two-layer MLP or a three-layer MLP, which represents linear mapping or nonlinear mapping, respectively. In this paper, we propose a new unified knowledge-based model structure which is expanded from [24]. This is done by expanding the mapping neural network from MLP to include an additional direct connection between the input layer and the output layer of the neural network. This structure is a composite two-layer and three-layer MLP, which allows the neural network to have explicit linear/nonlinear mapping. Thus, the proposed unified model has three mixed linear and nonlinear mapping neural networks that are the input mapping, frequency mapping, and output mapping. The structure of the proposed unified knowledge-based model is shown in Fig. 1.

We define the various symbols needed to represent various aspects of the mapping structure, modeling and the training. Let $\boldsymbol{f}_{\text{map1}}$, f_{map2}, and $\boldsymbol{f}_{\text{map3}}$ represent the mapping functions for input mapping, frequency mapping, and output mapping, respectively. Each mapping function is a neural network. Let H_1, H_2, and H_3 represent the number of hidden neurons in $\boldsymbol{f}_{\text{map1}}$, f_{map2}, and $\boldsymbol{f}_{\text{map3}}$, respectively. We define $\boldsymbol{u}_{\text{map1}}, \boldsymbol{u}_{\text{map2}}$, and $\boldsymbol{u}_{\text{map3}}$ as vectors of the weights for the proposed direct connections between the input neurons and output neurons in the input mapping $\boldsymbol{f}_{\text{map1}}$, frequency mapping f_{map2}, and output mapping $\boldsymbol{f}_{\text{map3}}$, respectively. Let $\boldsymbol{u} = [\boldsymbol{u}_{\text{map1}}^T\ \boldsymbol{u}_{\text{map2}}^T\ \boldsymbol{u}_{\text{map3}}^T]^T$ be defined as a vector containing all weights for the proposed direct connections between the input neurons and output neurons in all mapping neural networks. We define $\boldsymbol{t}_{\text{map1}}, \boldsymbol{t}_{\text{map2}}$, and $\boldsymbol{t}_{\text{map3}}$ as vectors of the weights between the input neurons and hidden neurons in $\boldsymbol{f}_{\text{map1}}$, f_{map2}, and $\boldsymbol{f}_{\text{map3}}$, respectively. We also define $\boldsymbol{v}_{\text{map1}}, \boldsymbol{v}_{\text{map2}}$, and $\boldsymbol{v}_{\text{map3}}$ as vectors of the weights between the hidden neurons and output neurons in $\boldsymbol{f}_{\text{map1}}$, f_{map2}, and $\boldsymbol{f}_{\text{map3}}$, respectively. Let $\boldsymbol{v} = [\boldsymbol{v}_{\text{map1}}^T\ \boldsymbol{v}_{\text{map2}}^T\ \boldsymbol{v}_{\text{map3}}^T]^T$ be defined as a vector containing all weights between the hidden neurons and output neurons in all mapping neural networks. Therefore, in our proposed unified model structure, the weights in $\boldsymbol{u}$ represent the linear mapping weights and the weights in $\boldsymbol{v}$ represent the nonlinear mapping weights.

We define $\boldsymbol{x}_E = [x_{E1}, x_{E2}, \ldots, x_{EN}]^T, \omega_E$ and $\boldsymbol{y}_E = [y_{E1}, y_{E2}, \ldots, y_{EM}]^T$ as the geometrical inputs, the frequency input, and the outputs of the empirical model, respectively. The empirical model is represented as

$$\boldsymbol{y}_E = \boldsymbol{f}_E\left(\left[\boldsymbol{x}_E^T\ \omega_E\right]^T, \boldsymbol{w}_E\right) \tag{1}$$

where $\boldsymbol{w}_E$ is a vector of parameters in the empirical model. Various empirical models have been developed in the past. They are computationally efficient but the accuracy is limited when matching new EM data. In other words, if we directly supply the geometrical parameter values $\boldsymbol{x}$ to the empirical model, the outputs of the empirical model may not sufficiently match the outputs $\boldsymbol{y}$ of the EM data, that is,

$$\boldsymbol{y} \neq \boldsymbol{f}_E([\boldsymbol{x}^T \omega]^T, \boldsymbol{w}_E). \tag{2}$$

This paper addresses the problem when the existing empirical model cannot fit the new EM data well. We use mappings to alter the relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$, therefore altering the model. The input mapping is used to modify the values of the geometrical parameters to a new set of values to be supplied to the empirical model. The purpose to modify (map) the values of the geometrical parameters is to make the subsequent response of the empirical model better match the EM simulation data. $\boldsymbol{x}$ represents the original values of the geometrical parameters and $\boldsymbol{x}_E$ represent the modified values

of the geometrical parameters to be supplied to the empirical model. Because the relationship between $\boldsymbol{x}$ and $\boldsymbol{x}_E$ is unknown, we use a neural network to represent this relationship, i.e., the input mapping function. In the proposed algorithm, the input mapping neural network, which has N input neurons and N output neurons, is defined as

$$\boldsymbol{x}_E = \boldsymbol{f}_{\text{map1}}(\boldsymbol{x}, \boldsymbol{w}_{\text{map1}}) \tag{3}$$

where $\boldsymbol{w}_{\text{map1}} = [\boldsymbol{u}_{\text{map1}}^T \ \boldsymbol{t}_{\text{map1}}^T \ \boldsymbol{v}_{\text{map1}}^T]^T$ represents the weights in the input mapping neural network $\boldsymbol{f}_{\text{map1}}$. The proposed algorithm determines the structure of the input mapping, whether linear or nonlinear, or no mapping. Here, we formulate the internal structure of the input mapping neural network to include linear and nonlinear mappings as follows. The ith output of the input mapping is computed as

$$f_{\text{map1},i} = \sum_{j=0}^{N} u_{\text{map1},ij} x_j + \sum_{k=0}^{H_1} (v_{\text{map1},ik} z_{\text{map1},k}), \quad i = 1, 2, \ldots, N \tag{4-1}$$

$$z_{\text{map1},k} = \begin{cases} \sigma\left(\sum_{j=0}^{N} t_{\text{map1},kj} x_j\right), & k \neq 0 \\ 1, & k = 0 \end{cases} \tag{4-2}$$

$$x_0 = 1 \tag{4-3}$$

where $z_{\text{map1},k}$ is defined as the output of the kth hidden neuron of the input mapping neural network. To accommodate the convenient representation of the bias parameters in neural networks, we assume the fictitious neurons in the input mapping as x_0 and $z_{\text{map1},0}$ [3]. $\sigma(\cdot)$ is the sigmoid function. $u_{\text{map1},ij}$, $v_{\text{map1},ik}$, and $t_{\text{map1},kj}$ represent the connection weight between the jth input neuron and the ith output neuron, the weight between the kth hidden neuron and the ith output neuron, and the weight between the jth input neuron and the kth hidden neuron of the input mapping neural network, respectively. $u_{\text{map1},ij}$ represents the linear mapping weight and $v_{\text{map1},ik}$ represents the nonlinear mapping weight in the input mapping neural network. In this way, the proposed input mapping can encompass linear mapping, nonlinear mapping or no mapping by one neural network.

The frequency mapping is usually used for those cases where a shift in frequency exists between the empirical model response and the EM data. The proposed algorithm determines the structure of the frequency mapping, whether linear or nonlinear, or no mapping. The internal structure of the frequency mapping neural network, which has one input neuron and one output neuron, is defined as

$$\begin{aligned} \omega_E &= f_{\text{map2}}(\omega, \boldsymbol{w}_{\text{map2}}) \\ &= f_{\text{map2}}\left(\omega, \left[\boldsymbol{u}_{\text{map2}}^T \ \boldsymbol{t}_{\text{map2}}^T \ \boldsymbol{v}_{\text{map2}}^T\right]^T\right) \\ &= \sum_{m=0}^{1} u_{\text{map2},1m} \omega_m + \sum_{n=0}^{H_2} (v_{\text{map2},1n} z_{\text{map2},n}) \end{aligned} \tag{5-1}$$

$$z_{\text{map2},n} = \begin{cases} \sigma\left(\sum_{m=0}^{1} t_{\text{map2},nm} \omega_m\right), & n \neq 0 \\ 1, & n = 0 \end{cases} \tag{5-2}$$

$$\omega_0 = 1 \tag{5-3}$$

where $\boldsymbol{w}_{\text{map2}} = [\boldsymbol{u}_{\text{map2}}^T \ \boldsymbol{t}_{\text{map2}}^T \ \boldsymbol{v}_{\text{map2}}^T]^T$ represents the weights in the frequency mapping neural network f_{map2}. $z_{\text{map2},n}$ is defined as the output of the nth hidden neuron of the frequency mapping neural network. In order to accommodate the convenient representation of the bias parameters in neural networks, we assume the fictitious neurons in the frequency mapping as ω_0 and $z_{\text{map2},0}$ [3]. $u_{\text{map2},1m}$, $v_{\text{map2},1n}$, and $t_{\text{map2},nm}$ represent the connection weight between the mth input neuron and the output neuron, the weight between the nth hidden neuron and the output neuron, and the weight between the mth input neuron and the nth hidden neuron of the frequency mapping neural network, respectively. $u_{\text{map2},1m}$ represents the linear mapping weight and $v_{\text{map2},1n}$ represents the nonlinear mapping weight in the frequency mapping neural network. Therefore, the proposed frequency mapping can encompass linear mapping, nonlinear mapping, or no mapping by one neural network.

The output mapping is used to enhance the robustness of the modeling process. In the proposed method, the output mapping neural network, which has M inputs and M outputs, is defined as

$$\boldsymbol{y} = \boldsymbol{f}_{\text{map3}}(\boldsymbol{y}_E, \boldsymbol{w}_{\text{map3}}) \tag{6}$$

where $\boldsymbol{w}_{\text{map3}} = [\boldsymbol{u}_{\text{map3}}^T \ \boldsymbol{t}_{\text{map3}}^T \ \boldsymbol{v}_{\text{map3}}^T]^T$ represents the weights in the output mapping neural network $\boldsymbol{f}_{\text{map3}}$. Similar to (4-1) and (5-1), here we formulate the internal structure of the output mapping neural network to include linear and nonlinear mappings as follows. The rth output of the output mapping is computed as

$$f_{\text{map3},r} = \sum_{p=0}^{M} u_{\text{map3},rp} y_{E,p} + \sum_{q=0}^{H_3} \left(v_{\text{map3},rq} z_{\text{map3},q}\right), \quad r = 1, 2, \ldots, M \tag{7-1}$$

$$z_{\text{map3},q} = \begin{cases} \sigma\left(\sum_{p=0}^{M} t_{\text{map3},qp} y_{E,p}\right), & q \neq 0 \\ 1, & q = 0 \end{cases} \tag{7-2}$$

$$y_{E,0} = 1 \tag{7-3}$$

where $z_{\text{map3},q}$ is defined as the output of the qth hidden neuron of the output mapping neural network. To accommodate the convenient representation of the bias parameters in neural networks, we assume the fictitious neurons in the output mapping as $y_{E,0}$ and $z_{\text{map3},0}$ [3]. $u_{\text{map3},rp}$, $v_{\text{map3},rq}$, and $t_{\text{map3},qp}$ represent the connection weight between the pth input neuron and the rth output neuron, the weight between the qth hidden neuron and the rth output neuron, and the weight between the pth input neuron and the qth hidden neuron of the output mapping neural network, respectively. $u_{\text{map3},rp}$ represents the linear mapping weight and $v_{\text{map3},rq}$ represents the nonlinear mapping weight in the output mapping neural network. In this way, the output mapping can encompass linear mapping, nonlinear mapping, or no mapping by one neural network.

We define $\boldsymbol{w} = [\boldsymbol{w}_{\text{map1}}^T \ \boldsymbol{w}_{\text{map2}}^T \ \boldsymbol{w}_{\text{map3}}^T]^T$ as a vector containing all the neural network weights that are treated as optimization variables during the training process. Combining all the mappings with the empirical model, the overall relationship

between $\boldsymbol{x}$ and $\boldsymbol{y}$, i.e., between the inputs and outputs of the overall model, can be represented as

$$\begin{aligned}\boldsymbol{y} &= \boldsymbol{f}_{\text{map3}}(\boldsymbol{y}_E, \boldsymbol{w}_{\text{map3}}) \\ &= \boldsymbol{f}_{\text{map3}}\left(\boldsymbol{f}_E\left(\left[\boldsymbol{x}_E^T\ \omega_E\right]^T, \boldsymbol{w}_E\right), \boldsymbol{w}_{\text{map3}}\right) \\ &= \boldsymbol{f}_{\text{map3}}\left(\boldsymbol{f}_E\left(\left[\boldsymbol{f}_{\text{map1}}^T(\boldsymbol{x}, \boldsymbol{w}_{\text{map1}})\, f_{\text{map2}}(\omega, \boldsymbol{w}_{\text{map2}})\right]^T, \right.\right. \\ &\qquad \left.\left. \boldsymbol{w}_E\right), \boldsymbol{w}_{\text{map3}}\right). \end{aligned} \tag{8}$$

This input and output relationship depends on not only the empirical model, but also the various mappings. Therefore, changing the mappings (including changing the mapping structure and changing the values of neural network weights in the mappings) will allow us to change the model.

This unified knowledge-based model includes all cases of mappings, which are no mapping, linear mapping, and nonlinear mapping for each of input mapping, frequency mapping, and output mapping. Usually, a modeling problem with a small range of geometrical parameters needs linear mappings and a modeling problem with a large range of geometrical parameters needs nonlinear mappings. However, a quantitative decision of when to use linear mapping or when to use nonlinear mapping is problem-dependent and is unknown in advance. The input mapping, frequency mapping, and output mapping are not necessarily all linear or all nonlinear. They can be different combinations of linear and nonlinear functions because they have different effects on the modeling behavior.

Existing automated model structure adaptation techniques for knowledge-based model development use a brute force sequential trial and error mechanism to try different combinations of SM structures and compare to determine the most suitable knowledge-based neural network structure. Here, we utilize the new unified knowledge-based model structure and propose a new training method to automatically determine the mapping structure in the final model.

III. Proposed Automated Model Generation Algorithm Using l_1 Optimization

A. Proposed Training Method With l_1 Optimization

In order to train the model, we generate training data. We define the training data as pairs of $(\boldsymbol{x}^{(a)}, \boldsymbol{d}^{(a)})$, $a = 1, 2, \ldots, N_A$, where $\boldsymbol{d}^{(a)}$ is a vector containing the desired outputs of the overall knowledge-based neural network model for the ath training sample $\boldsymbol{x}^{(a)}$, and N_A is the total number of training samples. For training purposes, the error function $E(\boldsymbol{w})$ in the existing knowledge-based neural network literature (see [3]) is denoted as

$$E(\boldsymbol{w}) = \sum_{a=1}^{N_A} \left(\frac{1}{2} \|\boldsymbol{y}(\boldsymbol{x}^{(a)}, \boldsymbol{w}) - \boldsymbol{d}^{(a)}\|^2 \right) \tag{9}$$

which represents the difference between the model outputs and the EM data outputs over all the training samples.

In the proposed method, we perform model training with l_1 optimization. The training process is divided into two stages. In the first stage, the proposed algorithm determines whether to use linear mapping or nonlinear mapping in various parts of the model. In the second stage, the algorithm determines whether a linear mapping, if exists, can be removed or not. After these two stages, the proposed method automatically produces the mapping structure of the knowledge-based model for the modeling problem.

For the first stage, we propose a new formulation of the error function for training. We add the l_1 norms of the neural network weights between hidden neurons and output neurons to the traditional training error function in (9). The new training error function in the first stage of our proposed training process is formulated as

$$\begin{aligned} E_{\text{train}}^{(1)}(\boldsymbol{w}) = &\sum_{a=1}^{N_A} \left(\frac{1}{2} \|\boldsymbol{y}(\boldsymbol{x}^{(a)}, \boldsymbol{w}) - \boldsymbol{d}^{(a)}\|^2 \right) \\ &+ \sum_{i=1}^{N} \sum_{k=0}^{H_1} \lambda_{\text{map1},ik}^{(1)} |v_{\text{map1},ik}| \\ &+ \sum_{n=0}^{H_2} \lambda_{\text{map2},1n}^{(1)} |v_{\text{map2},1n}| \\ &+ \sum_{r=1}^{M} \sum_{q=0}^{H_3} \lambda_{\text{map3},rq}^{(1)} |v_{\text{map3},rq}| \end{aligned} \tag{10}$$

where $\lambda_{\text{map1},ik}^{(1)}$, $\lambda_{\text{map2},1n}^{(1)}$ and $\lambda_{\text{map3},rq}^{(1)}$ are all nonnegative constants during the first stage training process. The proposed training problem of the first stage is to minimize the training error by adjusting $\boldsymbol{w}$ and force as many nonlinear mapping weights in $\boldsymbol{v}$ to zeros as possible.

The use of the l_1 norm as compared to the other l_p norms with $p > 1$ has the distinctive property that some large components of nonlinear mapping weight vector $\boldsymbol{v}$ can be ignored during the optimization process [26], [27], i.e., at the solution there may well be a few nonlinear mapping weights v's which are nonzeros while others are zeros. This means that, the important components of nonlinear mapping weight vector $\boldsymbol{v}$ can be automatically selected by the l_1 norm. The robustness of the l_1 optimization in dealing with large components of nonlinear mapping weight vector $\boldsymbol{v}$ is the result of a mathematical property related to the necessary conditions for optimality [28]. The solution of the first stage training is usually situated at a point where one or more of the v's equal zero while some large v's are in effect ignored completely by optimization process.

The formulation in (10) has two properties. The purpose of the first part in the error function is to optimize the model to match the training data. By adding the remaining parts to the error function, we penalize the error function for any large values of the nonlinear mapping weights in $\boldsymbol{v}$. Since the l_1 norm is used, one or a few large v's are still allowed. In this way, the model optimization can get good training accuracy as well as many v's to zeros. After the first stage training, the nonzero v's represent the selected nonlinear mapping function and nonlinear mapping structure. If some of the nonlinear mapping weights v's are nonzeros, the mapping is nonlinear mapping; if all nonlinear mapping weights v's are zeros, the nonlinear mapping can be deleted, and the remaining mapping structure is linear mapping. In this way, the proposed algorithm automatically detects and determines whether to use linear mapping or nonlinear mapping.

We have three mapping neural networks in our proposed knowledge-based model and each mapping neural network can be linear or nonlinear. If all these three mappings remain nonlinear mappings after the first stage training, the algorithm will stop. Otherwise, the algorithm proceeds to the second stage of the training process. The three mappings in the first stage solution can be all linear, or all nonlinear, or a mixture of some linear and some nonlinear. For the linear mapping, we propose to further retrain the neural network weights during the second stage of training process. For the nonlinear mapping, the neural network weights are set as constants during the second stage training. In the second stage of the training process, our proposed algorithm determines whether the linear mappings of the first stage training solution can be further simplified into unit mapping or not. Unit mapping means the values of the variables before mapping and after mapping are the same. If a mapping is unit mapping, it means this mapping is unnecessary and can be removed. Therefore, the second stage of the training process can determine whether a linear mapping is needed or not. The proposed l_1 training error function in the second stage is formulated as

$$\begin{aligned} E^{(2)}_{\text{train}}(\boldsymbol{u}) = &\sum_{a=1}^{N_A}\left(\frac{1}{2}\|\boldsymbol{y}(\boldsymbol{x}^{(a)},\boldsymbol{w})-\boldsymbol{d}^{(a)}\|^2\right) \\ &+\sum_{i=1}^{N}\sum_{j=0,j\neq i}^{N}\lambda^{(2)}_{\text{map1},ij}|u_{\text{map1},ij}| \\ &+\sum_{i=1}^{N}\lambda^{(2)}_{\text{map1},ii}|u_{\text{map1},ii}-1| \\ &+\lambda^{(2)}_{\text{map2},10}|u_{\text{map2},10}|+\lambda^{(2)}_{\text{map2},11}|u_{\text{map2},11}-1| \\ &+\sum_{r=1}^{M}\sum_{p=0,p\neq r}^{M}\lambda^{(2)}_{\text{map3},rp}|u_{\text{map3},rp}| \\ &+\sum_{r=1}^{M}\lambda^{(2)}_{\text{map3},rr}|u_{\text{map3},rr}-1| \end{aligned} \tag{11}$$

where $\lambda^{(2)}_{\text{map1},ij}$, $\lambda^{(2)}_{\text{map2},10}$, $\lambda^{(2)}_{\text{map2},11}$, and $\lambda^{(2)}_{\text{map3},rp}$ are all nonnegative constants during the second stage training process. $\lambda^{(2)}_{\text{map1},ij}$ are set to be zeros if the input mapping is nonlinear mapping. Similarly, $\lambda^{(2)}_{\text{map2},10}$ and $\lambda^{(2)}_{\text{map2},11}$ (or $\lambda^{(2)}_{\text{map3},rp}$) are set to be zeros if the frequency mapping (or the output mapping) is nonlinear mapping. In the second stage training solution, if

$$\begin{cases} u_{\text{map1},ij}=0, & i\neq j \\ u_{\text{map1},ij}=1, & i=j \end{cases} \tag{12}$$

the input mapping is unit mapping, that is

$$\boldsymbol{x}_E=\boldsymbol{x} \tag{13}$$

which means the input mapping is not needed in the final knowledge-based model. Similarly, if

$$\begin{cases} u_{\text{map2},10}=0 \\ u_{\text{map2},11}=1 \end{cases} \tag{14}$$

it means the frequency mapping is not needed in the final model. If

$$\begin{cases} u_{\text{map3},rp}=0, & r\neq p \\ u_{\text{map3},rp}=1, & r=p \end{cases} \tag{15}$$

the output mapping is unit mapping, which means the output mapping is not needed in the final model. Otherwise, the linear mappings are necessary in the final model. In this way, the proposed algorithm can automatically determine whether a mapping is needed or not.

B. Proposed Computation of the Error Derivatives in the Proposed Training Method With l_1 Optimization

During the training process, the derivatives of the error functions in (10) and (11) with respect to the weights in all the three mapping neural networks are needed. Due to the complexity of the proposed unified model covering all possibilities of mappings, the brute force derivation for derivatives of the proposed model structure is complicated. Here, we propose a simple and elegant approach based on the back propagation (BP) concept for MLP [29], extended to our unified knowledge-based structure and l_1 training error function. We define the error at the output of the overall model as

$$\Delta y_r^{(a)}=y_r^{(a)}-d_r^{(a)} \tag{16}$$

which represents the difference between the rth model output and the EM data output of the ath sample. Then the proposed BP algorithm will start from $\Delta y_r^{(a)}$ and propagate this error backward from the outputs of the output mapping neural network through the hidden layer of the output mapping to the inputs of the output mapping. Let $\Delta z_{\text{map3},q}$ be defined as the BP error back propagated from the output layer of the output mapping to the qth hidden neuron of the output mapping. The calculation of $\Delta z_{\text{map3},q}$ can be derived as

$$\Delta z_{\text{map3},q}=\sum_{r=1}^{M}\left[\Delta y_r^{(a)}\cdot v_{\text{map3},rq}\cdot z_{\text{map3},q}\cdot\left(1-z_{\text{map3},q}\right)\right] \tag{17}$$

where $\Delta y_r^{(a)}$ is the solution from (16).

The error BP continues from the output mapping to the empirical model. Let $\Delta y_{E,p}$ be defined as the BP error at the pth output of the empirical model. The calculation of $\Delta y_{E,p}$ can be derived as

$$\Delta y_{E,p}=\sum_{r=1}^{M}\left(\Delta y_r^{(a)}\cdot u_{\text{map3},rp}\right)+\sum_{q=0}^{H_3}\left(\Delta z_{\text{map3},q}\cdot t_{\text{map3},qp}\right) \tag{18}$$

where $\Delta y_r^{(a)}$ and $\Delta z_{\text{map3},q}$ are the solutions from (16) and (17), respectively.

Then the error BP continues from the empirical model to the output layer of the input mapping. Let $\Delta x_{E,i}$ be defined as the BP error at the ith output of the input mapping neural network, which can be derived as

$$\Delta x_{E,i}=\sum_{p=1}^{M}\left(\Delta y_{E,p}\cdot\frac{\partial y_{E,p}}{\partial x_{E,i}}\right) \tag{19}$$

where $\Delta y_{E,p}$ is the solution from (18). Let $\Delta z_{\text{map1},k}$ be defined as the BP error back propagated from the output layer of the input mapping to the kth hidden neuron of the input mapping. The calculation of $\Delta z_{\text{map1},k}$ can be derived as

$$\Delta z_{\text{map1},k} = \sum_{i=1}^{N} \left[\Delta x_{E,i} \cdot v_{\text{map1},ik} \cdot z_{\text{map1},k} \cdot \left(1 - z_{\text{map1},k}\right) \right] \tag{20}$$

where $\Delta x_{E,i}$ is the solution from (19).

The error BP also continues from the empirical model to the output layer of the frequency mapping. Let $\Delta \omega_E$ be defined as the BP error at the output of the frequency mapping neural network. It can be derived as

$$\Delta \omega_E = \sum_{p=1}^{M} \left(\Delta y_{E,p} \cdot \frac{\partial y_{E,p}}{\partial \omega_E} \right) \tag{21}$$

where $\Delta y_{E,p}$ is the solution from (18). Let $\Delta z_{\text{map2},n}$ be defined as the BP error back propagated from the output layer of the frequency mapping to the nth hidden neuron of the frequency mapping. The calculation of $\Delta z_{\text{map2},n}$ can be derived as

$$\Delta z_{\text{map2},n} = \Delta \omega_E \cdot v_{\text{map2},1n} \cdot z_{\text{map2},n} \cdot (1 - z_{\text{map2},n}) \tag{22}$$

where $\Delta \omega_E$ is the solution from (21).

After finishing the error BP process, the proposed training algorithm starts to calculate the derivatives of the training error with respect to the weights of the three mapping neural networks. Although the unified model structure is complicated, the final result of the sensitivity formula is surprisingly simple and elegant as described next. For the first stage of the training process, the error derivatives with respect to the weights in the output mapping can be derived as

$$\frac{\partial \left(E_{\text{train}}^{(1)}\right)^{(a)}}{\partial (u_{\text{map3},rp})} = \Delta y_r^{(a)} \cdot y_{E,p} \tag{23-1}$$

$$\frac{\partial \left(E_{\text{train}}^{(1)}\right)^{(a)}}{\partial (v_{\text{map3},rq})} = \Delta y_r^{(a)} \cdot z_{\text{map3},q} \pm \lambda_{\text{map3},rq}^{(1)} \tag{23-2}$$

$$\frac{\partial \left(E_{\text{train}}^{(1)}\right)^{(a)}}{\partial (t_{\text{map3},qp})} = \Delta z_{\text{map3},q} \cdot y_{E,p}. \tag{23-3}$$

If $v_{\text{map3},zrq} \geq 0$, the "±" symbol in (23-2) is replaced by the "+" symbol, otherwise is replaced by the "−" symbol. For the neural network weights in the input mapping, the error derivatives can be derived as

$$\frac{\partial \left(E_{\text{train}}^{(1)}\right)^{(a)}}{\partial \left(u_{\text{map1},ij}\right)} = \Delta x_{E,i} \cdot x_j \tag{24-1}$$

$$\frac{\partial \left(E_{\text{train}}^{(1)}\right)^{(a)}}{\partial \left(v_{\text{map1},ik}\right)} = \Delta x_{E,i} \cdot z_{\text{map1},k} \pm \lambda_{\text{map1},ik}^{(1)} \tag{24-2}$$

$$\frac{\partial \left(E_{\text{train}}^{(1)}\right)^{(a)}}{\partial \left(t_{\text{map1},kj}\right)} = \Delta z_{\text{map1},k} \cdot x_j. \tag{24-3}$$

If $v_{\text{map1},ik} \geq 0$, the "±" symbol in (24-2) is replaced by the "+" symbol, otherwise is replaced by the "–" symbol. For the neural network weights in the frequency mapping, the error derivatives can be derived as

$$\frac{\partial \left(E_{\text{train}}^{(1)}\right)^{(a)}}{\partial \left(u_{\text{map2},1m}\right)} = \Delta \omega_E \cdot \omega_m \tag{25-1}$$

$$\frac{\partial \left(E_{\text{train}}^{(1)}\right)^{(a)}}{\partial \left(v_{\text{map2},1n}\right)} = \Delta \omega_E \cdot z_{\text{map2},n} \pm \lambda_{\text{map2},1n}^{(1)} \tag{25-2}$$

$$\frac{\partial \left(E_{\text{train}}^{(1)}\right)^{(a)}}{\partial \left(t_{\text{map2},nm}\right)} = \Delta z_{\text{map2},n} \cdot \omega_m. \tag{25-3}$$

If $v_{\text{map2},1n} \geq 0$, the "±" symbol in (25-2) is replaced by the "+" symbol, otherwise is replaced by the "−" symbol.

For the second stage of the training process, the error derivatives with respect to the linear weights (i.e., $u_{\text{map3},rp}$) in the output mapping can be derived as

$$\frac{\partial \left(E_{\text{train}}^{(2)}\right)^{(a)}}{\partial \left(u_{\text{map3},rp}\right)} = \Delta y_r^{(a)} \cdot y_{E,p} \pm \lambda_{\text{map3},rp}^{(2)}. \tag{26}$$

If $u_{\text{map3},rp} \geq 0$, the "±" symbol in (26) is replaced by the "+" symbol, otherwise is replaced by the "−" symbol. For the linear neural network weights (i.e., $u_{\text{map1},ij}$) in the input mapping, the error derivatives can be derived as

$$\frac{\partial \left(E_{\text{train}}^{(2)}\right)^{(a)}}{\partial \left(u_{\text{map1},ij}\right)} = \Delta x_{E,i} \cdot x_j \pm \lambda_{\text{map1},ij}^{(2)}. \tag{27}$$

If $u_{\text{map1},ij} \geq 0$, the "±" symbol in (27) is replaced by the "+" symbol, otherwise it is replaced by the "−" symbol. For the linear neural network weights (i.e., $u_{\text{map2},1m}$) in the frequency mapping, the error derivatives can be derived as

$$\frac{\partial \left(E_{\text{train}}^{(2)}\right)^{(a)}}{\partial \left(u_{\text{map2},1m}\right)} = \Delta \omega_E \cdot \omega_m \pm \lambda_{\text{map2},1m}^{(2)}. \tag{28}$$

If $u_{\text{map2},1m} \geq 0$, the "±" symbol in (28) is replaced by the "+" symbol, otherwise it is replaced by the "−" symbol.

Using the above derivative information for training, the proposed algorithm proceeds to find a training solution that determines the mapping structure. In the proposed algorithm, we do not need to manually determine whether the mappings are linear or nonlinear, and whether an input, frequency, or output mapping is needed or not. The model structure can be automatically determined by training with l_1 optimization. The nonzero v's represent the selected nonlinear mapping function and nonlinear mapping structure. If the nonlinear mapping weights v's are nonzeros, the mapping is nonlinear mapping; if the nonlinear mapping weights v's are all zeros, the mapping is linear mapping. The nonzero linear mapping coefficients u's represent the selected linear mapping.

C. Simple Illustration Examples of the Mapping Type for One Mapping Neural Network

We take the input mapping as an example. If

$$\begin{cases} u_{\text{map1},ij} = 0, & i \neq j \\ u_{\text{map1},ij} = 1, & i = j \\ \boldsymbol{v}_{\text{map1}} = 0 \end{cases} \tag{29}$$

the input mapping function (4-1) is simplified as

$$f_{\text{map1},i} = x_j, \quad i = j = 1, 2, \ldots, N \tag{30}$$

which means the input mapping is unit mapping and there is no need for the input mapping in the final model.

If

$$\begin{cases} u_{\text{map1},ij} \neq 0, & i \neq j \\ \boldsymbol{v}_{\text{map1}} = \boldsymbol{0} \end{cases} \tag{31}$$

the input mapping function (4-1) is simplified as

$$f_{\text{map1},i} = \sum_{j=0}^{N} u_{\text{map1},ij} x_j, \quad i = 1, 2, \ldots, N \tag{32}$$

which means the input mapping $\boldsymbol{f}_{\text{map1}}$ is linear mapping.

If

$$\boldsymbol{v}_{\text{map1}} \neq \boldsymbol{0} \tag{33}$$

the input mapping function $\boldsymbol{f}_{\text{map1}}$ is nonlinear mapping.

The determinations for the frequency mapping and the output mapping neural networks are similar to that for the input mapping. In this way, the proposed algorithm uses l_1 optimization to automatically obtain a most suitable knowledge-based model.

D. Simple Illustration Examples of the Overall Model Structure Including Various Mappings

In this section, we show several examples of different model structures to illustrate that the final model can be various combinations of mapping neural networks after using l_1 optimization to obtain the knowledge-based model.

1) Model Structure Example 1: If after l_1 optimization, the nonlinear input mapping weights $\boldsymbol{v}_{\text{map1}} \neq \boldsymbol{0}$, the nonlinear frequency mapping weights $\boldsymbol{v}_{\text{map2}} = \boldsymbol{0}$, the linear frequency mapping weights $u_{\text{map2},10} = 0, u_{\text{map2},11} = 1$, the nonlinear output mapping weights $\boldsymbol{v}_{\text{map3}} = \boldsymbol{0}$, and the linear output mapping weights $u_{\text{map3},rp} = 0, (r \neq p)$, $u_{\text{map3},rp} = 1, (r = p)$, (8) will be simplified as

$$\boldsymbol{y} = \boldsymbol{f}_E\left(\left[\boldsymbol{f}_{\text{map1}}^T(\boldsymbol{x}, \boldsymbol{w}_{\text{map1}})\ \omega\right]^T, \boldsymbol{w}_E\right) \tag{34}$$

which means the final knowledge-based model has only a nonlinear input mapping. There is no need for frequency mapping and no need for output mapping in the final model.

2) Model Structure Example 2: If the nonlinear input mapping weights $\boldsymbol{v}_{\text{map1}} \neq \boldsymbol{0}$, the nonlinear frequency mapping weights $\boldsymbol{v}_{\text{map2}} \neq \boldsymbol{0}$, the nonlinear output mapping weights $\boldsymbol{v}_{\text{map3}} = \boldsymbol{0}$, and the linear output mapping weights $u_{\text{map3},rp} = 0, (r \neq p)$, $u_{\text{map3},rp} = 1, (r = p)$, the final model will have a nonlinear input mapping and a nonlinear frequency mapping. There is no need for output mapping in the final model. Then (8) will be simplified as

$$\boldsymbol{y} = \boldsymbol{f}_E\left(\left[\boldsymbol{f}_{\text{map1}}^T(\boldsymbol{x}, \boldsymbol{w}_{\text{map1}})\ f_{\text{map2}}(\omega, \boldsymbol{w}_{\text{map2}})\right]^T, \boldsymbol{w}_E\right). \tag{35}$$

3) Model Structure Example 3: If the nonlinear input mapping weights $\boldsymbol{v}_{\text{map1}} \neq \boldsymbol{0}$, the nonlinear frequency mapping weights $\boldsymbol{v}_{\text{map2}} = \boldsymbol{0}$, the linear frequency mapping weights $u_{\text{map2},10} \neq 0, u_{\text{map2},11} \neq 1$, and the nonlinear output mapping weights $\boldsymbol{v}_{\text{map3}} \neq \boldsymbol{0}$, the final model will contain a nonlinear input mapping, a linear frequency mapping, and a nonlinear output mapping. Then the model will be

$$\boldsymbol{y} = \boldsymbol{f}_{\text{map3}}\left(\boldsymbol{f}_E\left(\left[\boldsymbol{f}_{\text{map1}}^T(\boldsymbol{x}, \boldsymbol{w}_{\text{map1}})\ f_{\text{map2}}(\omega, \boldsymbol{w}_{\text{map2}})\right]^T, \boldsymbol{w}_E\right), \boldsymbol{w}_{\text{map3}}\right). \tag{36}$$

Using l_1 optimization, the proposed method can automatically determine the combination of input mapping, frequency mapping, and output mapping. Each mapping can be linear or nonlinear mapping or no mapping.

E. Proposed Automated Modeling Algorithm

The proposed unified automated modeling algorithm uses the training method with l_1 optimization to train a unified knowledge-based model structure, forcing the model to determine the type and topology of SM structure automatically. The testing error $E_{\text{test}}(\boldsymbol{w}^*)$ is defined as

$$E_{\text{test}}(\boldsymbol{w}^*) = \sum_{b=1}^{N_B}\left(\frac{1}{2}\|\boldsymbol{y}(\boldsymbol{x}^{(b)}, \boldsymbol{w}^*) - \boldsymbol{d}^{(b)}\|^2\right) \tag{37}$$

where $\boldsymbol{w}^*$ is the optimal solution after the training process containing the weight parameters of all the mapping neural networks. $\boldsymbol{d}^{(b)}$ represents the testing data output of the bth input sample $\boldsymbol{x}^{(b)}$ and N_B represents the number of testing data.

We define E_d as the user-defined threshold. During the first stage of the training process, once $E_{\text{test}} \leq E_d$ is detected, the algorithm checks the values of the trained nonlinear mapping weights $\boldsymbol{v}$ in all the three mapping neural networks. If all the three mappings are nonlinear mappings, the algorithm stops. Otherwise, the algorithm proceeds to the second stage of the training process to determine whether the linear mapping is needed or not. The algorithm will retrain the weights in the linear mapping until $E_{\text{test}} \leq E_d$ is achieved again.

The proposed algorithm can be summarized as follows.

- Step 1) Perform EM simulation to generate training and testing data using designed of experiments (DoE) sampling method [30].
- Step 2) Optimize the existing empirical model to make the model approximate the training data as much as possible.
- Step 3) Test the optimized empirical model. If the empirical model satisfies the accuracy requirement, which means the empirical model is good enough, stop the modeling process. Else, we add mappings to the empirical model to improve the model in subsequent steps.
- Step 4) Create initial mapping neural networks $\boldsymbol{f}_{\text{map1}}$, f_{map2}, and $\boldsymbol{f}_{\text{map3}}$ with five hidden neurons in each neural network, and initialize the weight parameters of the mapping neural networks.

Step 5) Perform the first stage training with l_1 optimization, using the training error function in (10) to determine whether the mappings are linear or nonlinear.

Step 6) Test the model. If $E_{\text{train}} > E_d$, the model is detected as under-learning [23]; add one more hidden neuron and go back to Step 5). Else if $E_{\text{train}} \leq E_d$ and $E_{\text{test}} > E_d$, the model is detected as over-learning [23]; delete one hidden neuron in the nonlinear neural networks and go back to Step 5). Else, go to Step 7).

Step 7) If $\boldsymbol{v}_{\text{map1}} \neq \mathbf{0}$ and $\boldsymbol{v}_{\text{map2}} \neq \mathbf{0}$ and $\boldsymbol{v}_{\text{map3}} \neq \mathbf{0}$, all mappings are nonlinear and all mapping weights in the neural networks are determined. Stop the modeling process. Else, at least one mapping is linear. Set the neural network weights in the nonlinear mappings as constants, and leave the neural network weights in the linear mappings as variables. Perform the second stage training with l_1 optimization until $E_{\text{test}} \leq E_d$, using the training error function in (11).

Step 8) If one or more linear mappings are unit mapping, which means the linear mappings are unnecessary, delete the unnecessary linear mappings and perform refinement training to fine-tune the overall model until $E_{\text{test}} \leq E_d$. Then stop the modeling process. Else, the linear mappings are needed in the final model. Stop the modeling process.

Fig. 2 shows the flow diagram of our proposed unified automated modeling algorithm using knowledge-based neural network and l_1 optimization.

F. Discussion

$\lambda^{(1)}_{\text{map1},ik}$, $\lambda^{(1)}_{\text{map2},1n}$, $\lambda^{(1)}_{\text{map3},rq}$ in (10) and $\lambda^{(2)}_{\text{map1},ij}$, $\lambda^{(2)}_{\text{map2},10}$, $\lambda^{(2)}_{\text{map2},11}$, $\lambda^{(2)}_{\text{map3},rp}$ in (11) are user-defined factors. The values of these factors are either zero or nonzero. Zero λ's mean the l_1 norms of the corresponding parameters in the mapping neural network are not included in the training error functions. Nonzero λ's mean that large values in the corresponding parameters of the mapping neural network are penalized using l_1 optimization. For nonzeroλ's, we use 10 as the default value in this paper. In (10), if the values of the λ's are large, the training process will emphasize more on forcing as many v's to zeros as possible, while emphasize less on optimizing the model to match the training data. On the other hand, if the values of the λ's in (10) are small but not equal to zero, the training error of the model will tend to be small but we cannot guarantee that the values of v's in the model are as small as possible. The λ's in (11) work in a similar way as those in (10).

An important difference between the proposed method and the previous automated modeling method [25] is that our algorithm finds the best knowledge-based models while the previous method produces pure neural network models. Knowledge-based models have better extrapolation capability than pure neural network models as originally demonstrated in [12]. Using the knowledge-based model formulation in this paper, the proposed training method enables the extrapolation capability of the models.

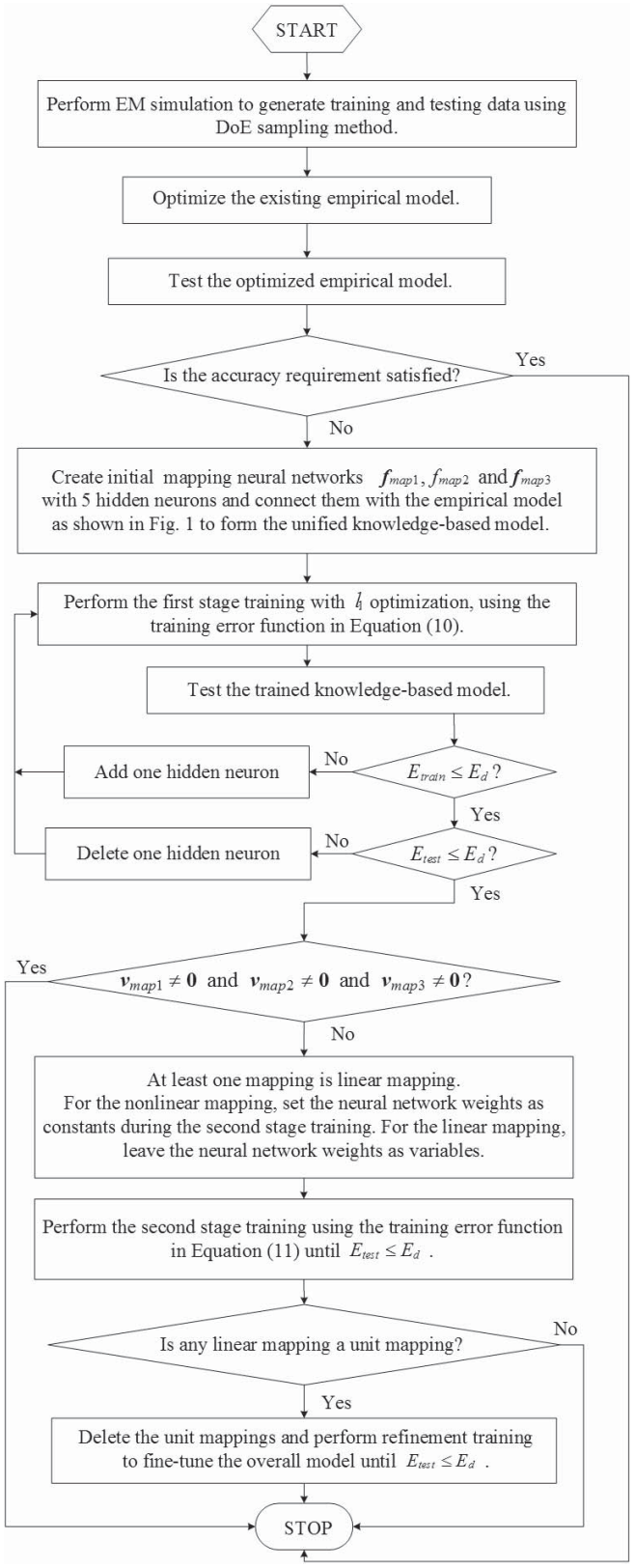

Fig. 2. Block diagram of the proposed unified automated knowledge-based neural network modeling method using l_1 optimization.

This paper focuses on proposing a computational method for parametric modeling to represent the change in EM behavior versus the change in geometrical parameters. We perform validation using independent EM data that are never used in

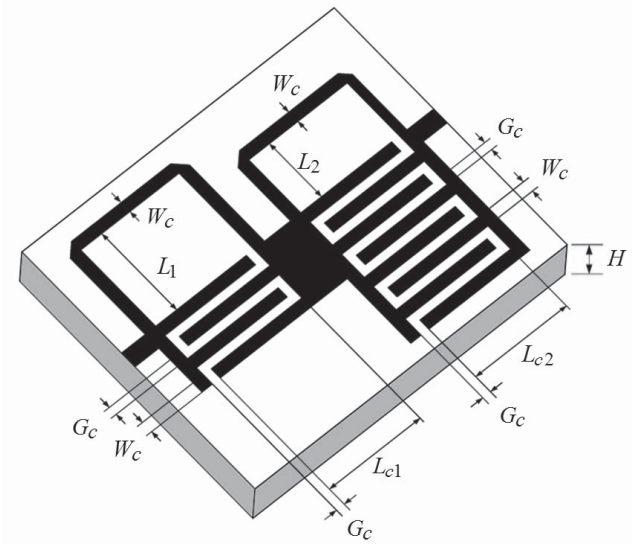

Fig. 3. Low-pass elliptic microstrip filter example: EM simulation for data generation.

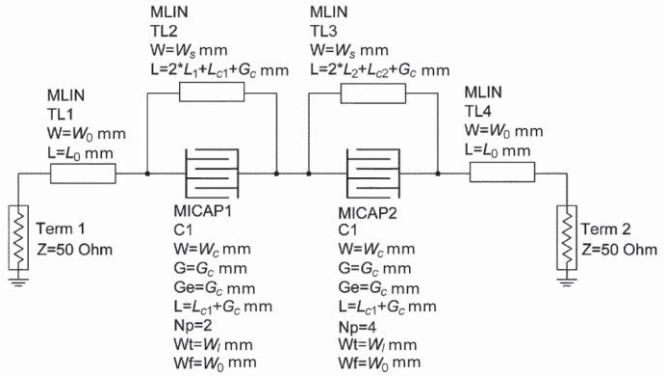

Fig. 4. Low-pass elliptic microstrip filter example: the empirical model.

training of the model. For validation of EM simulation data including measurement validation, we refer to existing works such as [31] and [32].

IV. Examples

A. Parametric Modeling of Two-Section Low-Pass Elliptic Microstrip Filter

In this example, we develop a knowledge-based parametric model for a two-section low-pass elliptic microstrip filter [33]–[35], as shown in Fig. 3. The training and testing data are obtained from EM simulator *CST Microwave Studio* using DoE sampling method [30]. In Fig. 3, H is the thickness of the alumina substrate and ε_r is the dielectric constant of the substrate. $H = 0.508$ mm and $\varepsilon_r = 9.4$. The geometrical input parameters of the model are $\boldsymbol{x} = [L_1 L_2 L_{c1} L_{c2} W_c G_c]^T$, chosen based on the sensitivity data of the above low-pass elliptic filter example. The model output is the magnitude of S_{21}. The existing empirical model is the equivalent circuit for the low-pass filter using simple transmission lines, which is shown in Fig. 4 [10], [20]. We build our empirical model using formulas based on *Keysight Advanced Design System (ADS)* [34], [36]. After optimization, the empirical model still cannot satisfy the accuracy requirement, therefore we proceed to add mappings using our proposed method. *NeuroModelerPlus* software is used to drive EM simulators for data generation, program the empirical model, implement unified mapping structures, and perform the knowledge-based model training and testing. The new training error functions in (10) and (11) are implemented through Huber functions in *NeuroModelerPlus.* The values of $\lambda^{(1)}_{\text{map1},ik}$, $\lambda^{(1)}_{\text{map2},1n}$, and $\lambda^{(1)}_{\text{map3},rq}$ in (10) are set to be 10.

For comparison purposes, we develop knowledge-based models for two different cases. The range of the geometrical parameters for Case 1 is small, while the range of the geometrical parameters for Case 2 is large. These two user-desired modeling ranges are shown in Table I. We set the testing error threshold for this modeling example as 2%.

TABLE I
Training Data and Testing Data for Parametric Modeling of the Low-Pass Filter Example

Input	Training Data		Testing Data	
	Min	Max	Min	Max
	Case 1			
L_1 (mm)	1.14	1.24	1.15	1.23
L_2 (mm)	3.3	3.5	3.31	3.49
L_{c1} (mm)	4.06	4.26	4.07	4.25
L_{c2} (mm)	1.14	1.24	1.15	1.23
W_c (mm)	0.16	0.2	0.165	0.195
G_c (mm)	0.05	0.07	0.052	0.069
Freq (GHz)	1	4	1	4
	Case 2			
L_1 (mm)	1.02	1.27	1.03	1.26
L_2 (mm)	3.18	3.68	3.2	3.65
L_{c1} (mm)	3.94	4.44	3.96	4.42
L_{c2} (mm)	1.0	1.27	1.02	1.25
W_c (mm)	0.15	0.25	0.16	0.24
G_c (mm)	0.05	0.09	0.053	0.085
Freq (GHz)	1	4	1	4

TABLE II
Comparison of the Modeling Results for Low-Pass Filter With Two Different Modeling Ranges

	Case 1	Case 2
Mapping Structure of the Final Model	Linear input mapping	Nonlinear input mapping + Nonlinear frequency mapping
No. of Training Data	81*101	81*101
No. of Testing Data	64*101	64*101
Training Error	1.89%	1.63%
Testing Error	**1.92%**	**1.97%**

The modeling results are listed in Table II and shown in Fig. 5. For Case 1, after using l_1 optimization, a linear input mapping is good enough to obtain an accurate knowledge-based model with about 2% testing error. The values of $\lambda^{(2)}_{\text{map1},ij}$, $\lambda^{(2)}_{\text{map2},10}$, $\lambda^{(2)}_{\text{map2},11}$, and $\lambda^{(2)}_{\text{map3},rp}$ in (11) are

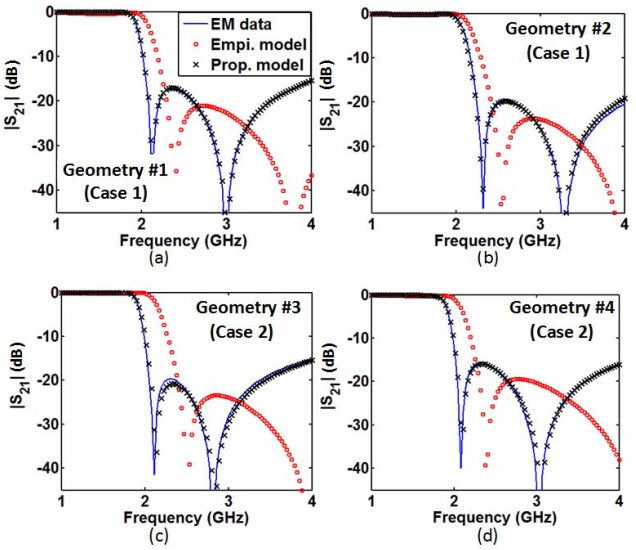

Fig. 5. Low-pass elliptic microstrip filter example: modeling results at four different geometrical values (a) $\boldsymbol{x} = [1.15, 3.49, 4.25, 1.16, 0.18, 0.06]^T$, (b) $\boldsymbol{x} = [1.23, 3.47, 4.08, 1.17, 0.19, 0.069]^T$, (c) $\boldsymbol{x} = [1.03, 3.52, 4.29, 1.22, 0.23, 0.058]^T$, and (d) $\boldsymbol{x} = [1.13, 3.46, 4.35, 1.10, 0.21, 0.06]^T$. The solid line, "o", and "x" in the figures represent the EM simulation data, the empirical model response and the proposed model response, respectively.

equal to 10. All nonlinear mapping weights $\boldsymbol{v}_{\text{map1}}, \boldsymbol{v}_{\text{map2}}$ and $\boldsymbol{v}_{\text{map3}}$ in mapping neural networks are almost zeros in the final knowledge-based model, therefore all mappings in the final model are linear mappings. Besides, the frequency mapping and output mapping are unit mapping, which means frequency mapping and output mapping are not needed. This result is consistent with our observation that since the range of the modeling space is small, a linear mapping is sufficient.

For Case 2, we also perform l_1 optimization during the model development. After l_1 optimization, the nonlinear mapping weights $\boldsymbol{v}_{\text{map1}}$ and $\boldsymbol{v}_{\text{map2}}$ of the input mapping and frequency mapping are nonzeros and the nonlinear mapping weights $\boldsymbol{v}_{\text{map3}}$ of the output mapping are zeros. The values of $\lambda^{(2)}_{\text{map1},ij}$, $\lambda^{(2)}_{\text{map2},10}$, and $\lambda^{(2)}_{\text{map2},11}$ in (11) are equal to 0, while the values of $\lambda^{(2)}_{\text{map3},rp}$ in (11) are equal to 10. Besides, the output mapping is unit mapping. This modeling result means the proposed algorithm with l_1 optimization automatically chooses a nonlinear input mapping and a nonlinear frequency mapping to map the difference between the empirical model and EM simulation. The output mapping is not needed. For Case 2, we have also generated another set of testing data, which is beyond the training range to test the proposed model, i.e., L_1 : 0.96–1.32 mm, L_2 : 3.05–3.81 mm, L_{c1} : 3.81–4.32 mm, L_{c2} : 0.96–1.32 mm, W_c : 0.12–0.28 mm, and G_c : 0.045–0.096 mm. The testing error is 2.35% which means the proposed model also has reasonable accuracy beyond the training range because of the embedded knowledge in the model. From these two modeling examples in Cases 1 and 2, it is demonstrated that the proposed algorithm uses l_1 optimization to determine whether to use linear mapping or nonlinear mapping for the specific modeling problem and determine the types of mapping structures automatically.

TABLE III
COMPARISON BETWEEN THE MODEL FROM THE PROPOSED AUTOMATED MODELING METHOD AND THE MODELS FROM EXISTING BRUTE FORCE MODELING ALGORITHMS WITH DIFFERENT COMBINATIONS OF MAPPINGS FOR THE LOW-PASS FILTER MODELING PROBLEM

Model Structure	Testing Error	
	Case 1	Case 2
Optimal Model Structure by Proposed Method : **Empirical Model + Linear Input Mapping**	**1.92%**	/
Optimal Model Structure by Proposed Method : **Empirical Model + Nonlinear Input Mapping + Nonlinear Frequency Mapping**	/	**1.97%**
Empirical Model Alone	16.43%	14.33%
Empirical Model + Linear Input Mapping	1.92%	2.38%
Empirical Model + Nonlinear Input Mapping	1.88%	2.04%
Empirical Model + Linear Output Mapping	14.95%	13.80%
Empirical Model + Nonlinear Output Mapping	7.73%	9.10%
Empirical Model + Linear Input Mapping + Linear Frequency Mapping + Linear Output Mapping	1.89%	2.17%
Empirical Model + Nonlinear Input Mapping + Nonlinear Frequency Mapping + Nonlinear Output Mapping	1.85%	1.96%

We compare the modeling results using existing brute force knowledge-based modeling algorithm and the proposed automated model generation algorithm, shown in Table III. In the brute force knowledge-based modeling method, we have to try different mapping structures in order to choose an accurate knowledge-based model. Here, Table III shows only some examples of the trials in the existing brute force modeling algorithm. In reality, the number of the possible combinations of the mappings in a knowledge-based model is more than that listed in Table III. We also compare the modeling time between the existing method in [24] and our proposed method, as shown in Table IV. The proposed method increases the modeling efficiency and develops the model in a shorter time than the brute force modeling approaches.

To further compare between the model produced by the proposed method and the fixed structured model, we consider a fixed model structure where all neural network mappings are nonlinear (refer to as "three nonlinear mapping structure"). We use less training data to develop models for Cases 1 and 2. The results are shown in Table V. The testing error of the

TABLE IV
COMPARISON OF THE MODELING TIME FOR LOW-PASS FILTER USING THE EXISTING METHOD IN [24] AND THE PROPOSED METHOD

	Case 1	Case 2
Existing Method in [24]	4.5 h	4.3 h
Proposed Method	3.3 h	3.35 h

TABLE V
COMPARISON OF THE MODELING RESULTS FOR LOW-PASS FILTER USING THE THREE NONLINEAR MAPPING STRUCTURE AND THE PROPOSED METHOD WITH TWO DIFFERENT MODELING RANGES

	Case 1		Case 2	
Mapping Structure of the Final Model	3 nonlinear mappings (fixed)	Linear input mapping (proposed)	3 nonlinear mappings (fixed)	Nonlinear input mapping (proposed)
No. of Training Data	49*101		49*101	
No. of Testing Data	64*101		64*101	
Training Error	1.21%	1.45%	1.87%	1.74%
Testing Error	2.82%	1.92%	2.43%	1.98%

model with three nonlinear mappings is larger than that of the proposed model, and cannot satisfy the user-desired accuracy. In this situation, the status of training the model with three nonlinear mappings is over-learning. More training data are needed to obtain an accurate model satisfying the user-desired accuracy. Because the proposed algorithm with l_1 optimization forces the model to be as simple as possible while preserving accuracy, less data are needed by the proposed method to develop the knowledge-based model. Therefore, the proposed technique is more efficient than simply choosing three nonlinear mappings when the number of training data is limited. In addition, the model produced by the proposed method is simpler and more compact than the model with three nonlinear mappings. The resulting compact model is easier to use and easier to be incorporated into circuit simulators.

B. Parametric Modeling of Bandpass HTS Microstrip Filter

Consider the parametric modeling of an HTS quarter-wave parallel coupled-line microstrip filter [16], [17], [35], illustrated in Fig. 6. In the figure, $L_1, L_2,$ 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. L_0 is the length of the input and output transmission line feeding the coupled line filter. The width $W = 0.635$ mm is assumed to be the same for all the sections. A lanthanum-aluminate substrate with thickness $H = 0.508$ mm and dielectric constant $\varepsilon_r = 23.425$ is used.

The model input variables $\boldsymbol{x} = [L_1\ L_2\ L_3\ S_1\ S_2\ S_3]^T$ for this example are chosen based on the sensitivity data and the model output is the magnitude of S_{21}. The testing error criteria for this modeling example is 4%. Training and testing data

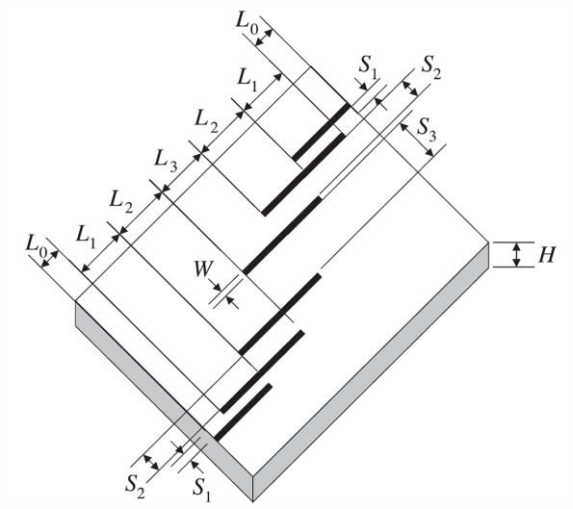

Fig. 6. Bandpass HTS microstrip filter example: EM simulation for data generation.

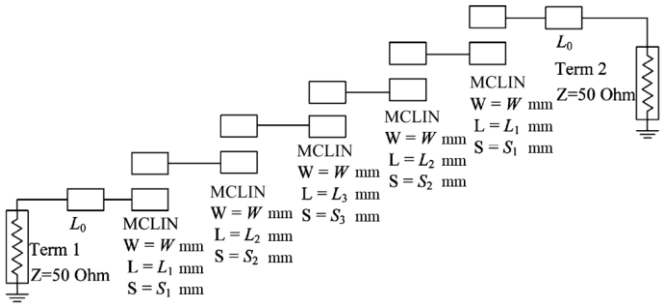

Fig. 7. Bandpass HTS microstrip filter example: the empirical model.

generation is performed by *CST Microwave Studio* EM simulator using DoE sampling method [30]. We build the empirical model using formulas based on *Keysight ADS* [36], [37], shown in Fig. 7 [17]. After the optimization of the empirical model, the error is still beyond our threshold. Therefore, we proceed to add mappings using our proposed method. The values of $\lambda^{(1)}_{\text{map2},1n}$, and $\lambda^{(1)}_{\text{map3},rq}$ in (10) are set to be 10.

We develop knowledge-based bandpass filter models for two different sets of geometrical parameter ranges, shown in Table VI. For Case 1, the range of the geometrical parameters is small, while the range of the geometrical parameters for Case 2 is large. The modeling results are listed in Table VII and Fig. 8 shows the comparison of the empirical model, the proposed knowledge-based model, and the EM simulation at four different sets of geometrical values.

For Case 1, the modeling range is relatively small. We perform l_1 optimization to develop the model. After l_1 optimization, the nonlinear mapping weights $\boldsymbol{\nu}_{\text{map1}}, \boldsymbol{\nu}_{\text{map2}}$, and $\boldsymbol{\nu}_{\text{map3}}$ are almost zeros, which means all mappings are linear mapping. The values of$\lambda^{(2)}_{\text{map1},ij}$, $\lambda^{(2)}_{\text{map2},10}$, $\lambda^{(2)}_{\text{map2},11}$, and $\lambda^{(2)}_{\text{map3},rp}$ in (11) are equal to 10. Besides, the linear frequency mapping and the linear output mapping are unit mapping, which means these two mappings are not needed. Therefore, the proposed algorithm with l_1 optimization uses only a linear input mapping to obtain a knowledge-based

TABLE VI

TRAINING DATA AND TESTING DATA FOR PARAMETRIC MODELING OF THE BANDPASS FILTER EXAMPLE

Input	Training Data		Testing Data	
	Min	Max	Min	Max
		Case 1		
L_1 (mm)	4.56	4.78	4.58	4.76
L_2 (mm)	4.82	5.03	4.84	5.01
L_3 (mm)	4.57	4.78	4.58	4.76
S_1 (mm)	0.4	0.6	0.41	0.59
S_2 (mm)	2.03	2.24	2.04	2.22
S_3 (mm)	2.28	2.48	2.3	2.47
Freq (GHz)	3.8	4.2	3.8	4.2
		Case 2		
L_1 (mm)	4.57	4.98	4.6	4.95
L_2 (mm)	4.82	5.23	4.85	5.2
L_3 (mm)	4.57	4.98	4.6	4.95
S_1 (mm)	0.46	0.54	0.47	0.53
S_2 (mm)	1.78	1.98	1.79	1.97
S_3 (mm)	2.03	2.24	2.04	2.22
Freq (GHz)	3.8	4.2	3.8	4.2

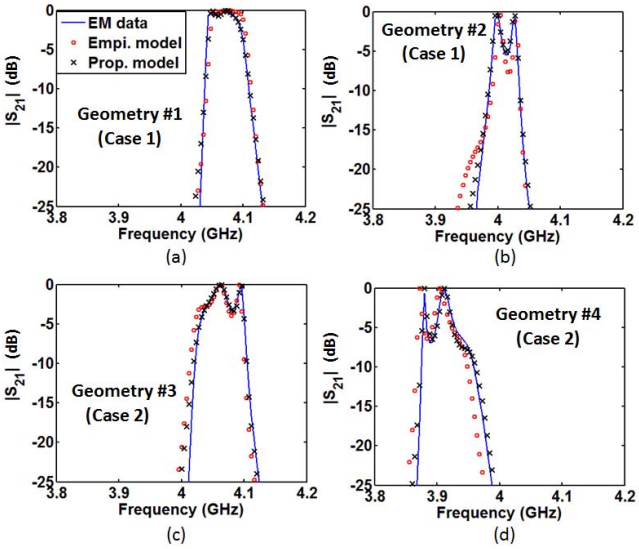

Fig. 8. Bandpass HTS microstrip filter example: modeling results at four different geometrical values (a) $\boldsymbol{x} = [4.58, 4.86, 4.61, 0.47, 2.12, 2.4]^T$, (b) $\boldsymbol{x} = [4.76, 5.0, 4.64, 0.42, 2.12, 2.45]^T$, (c) $\boldsymbol{x} = [4.65, 4.9, 4.6, 0.5, 1.97, 2.1]^T$, and (d) $\boldsymbol{x} = [4.8, 5.0, 4.9, 0.47, 1.96, 2.12]^T$. The solid line, "o", and "x" in the figures represent the EM simulation data, the empirical model response and the proposed model response, respectively.

model that meets the user-desired accuracy. This final model structure from l_1 optimization is consistent with observation that a linear mapping is sufficient for the modeling problem with a relatively small modeling range.

For Case 2, after l_1 optimization, the nonlinear weights $\boldsymbol{v}_{\text{map2}}$ of the frequency mapping are all zeros while $\boldsymbol{v}_{\text{map1}}$ and $\boldsymbol{v}_{\text{map3}}$ in the input mapping and output mapping are nonzeros. Therefore, the final model contains a nonlinear input mapping, a linear frequency mapping, and a nonlinear output mapping. The values of $\lambda^{(2)}_{\text{map1},ij}$ and $\lambda^{(2)}_{\text{map3},rp}$ in (11) are equal to 0,

TABLE VII

COMPARISON OF THE MODELING RESULTS FOR BANDPASS FILTER WITH TWO DIFFERENT MODELING RANGES

	Case 1	Case 2
Mapping Structure of the Final Model	Linear input mapping	Nonlinear input mapping + Linear frequency mapping + Nonlinear output mapping
No. of Training Data	81*101	81*101
No. of Testing Data	64*101	64*101
Training Error	4.13%	3.87%
Testing Error	**3.96%**	**3.87%**

TABLE VIII

COMPARISON BETWEEN THE MODEL FROM THE PROPOSED AUTOMATED MODELING METHOD AND THE MODELS FROM EXISTING BRUTE FORCE MODELING ALGORITHMS WITH DIFFERENT COMBINATIONS OF MAPPINGS FOR THE BANDPASS FILTER MODELING PROBLEM

Model Structure	Testing Error	
	Case 1	Case 2
Optimal Model Structure by Proposed Method : **Empirical Model + Linear Input Mapping**	**3.96%**	/
Optimal Model Structure by Proposed Method : **Empirical Model + Nonlinear Input Mapping + Linear Frequency Mapping + Nonlinear Output Mapping**	/	**3.87%**
Empirical Model Alone	9.43%	12.11%
Empirical Model + Linear Input Mapping	3.96%	9.16%
Empirical Model + Nonlinear Input Mapping	4.06%	4.66%
Empirical Model + Linear Output Mapping	9.25%	11.91%
Empirical Model + Nonlinear Output Mapping	9.02%	11.59%
Empirical Model + Linear Input Mapping + Linear Frequency Mapping + Linear Output Mapping	3.94%	6.24%
Empirical Model + Nonlinear Input Mapping + Nonlinear Frequency Mapping + Nonlinear Output Mapping	3.87%	3.48%

TABLE IX
COMPARISON OF THE MODELING TIME FOR BANDPASS FILTER USING THE EXISTING METHOD IN [24] AND THE PROPOSED METHOD

	Case 1	Case 2
Existing Method in [24]	6.5 h	6.75 h
Proposed Method	4.32 h	5.03 h

while the values of $\lambda^{(2)}_{\text{map2,10}}$ and $\lambda^{(2)}_{\text{map2,11}}$ in (11) are equal to 10. For Case 2, we have also generated another set of testing data, which is beyond the training range to test the proposed model, i.e., L_1 : 4.47–5.08 mm, L_2 : 4.72–5.33 mm, L_3 : 4.47–5.08 mm, S_1 : 0.44–0.58 mm, S_2 : 1.72–2.03 mm, and S_3 : 1.98–2.29 mm. The testing error is 5.54%, which means the extrapolation capability of the proposed model is reasonable because of the embedded knowledge in the model. The model structure in Case 2 is more complicated than that in Case 1, which is consistent with the fact that the modeling range is larger and the nonlinearity of the SM is stronger for Case 2 than for Case 1. The comparison of the modeling results between the existing brute force knowledge-based modeling methods and the proposed automated modeling algorithm is listed in Table VIII. In reality, the number of the possible combinations of the mappings in a knowledge-based model is more than that listed in Table VIII. We also compare the modeling time between the existing method in [24] and our proposed method, as shown in Table IX.

The modeling results illustrate that our proposed automated knowledge-based model generation algorithm with l_1 optimization has the ability to distinguish the linearity of specific modeling example. For different modeling examples, it can develop knowledge-based models with different mapping structures. The final model is forced to be as compact as possible by l_1 optimization property to force as many weights to zeros as possible. In addition, the first part in the l_1 error function makes the final model satisfy the user-required accuracy at the same time. Our proposed algorithm increases the modeling efficiency and reduces the modeling time compared with the existing method.

C. Parametric Modeling of Bandstop Microstrip Filter With Open Stubs

The third example is the parametric model development for a bandstop microstrip filter with quarter-wave resonant open stubs [15], [17], [35], illustrated in Fig. 9. In the figure, L_1 and L_2 are the open stub lengths, and W_1 and W_2 are the open stub widths. L_0 is the interconnecting transmission line length between the two open stubs. W_0 is the width of a 50 Ω feeding line and $W_0 = 0.635$ mm. An alumina substrate with thickness of $H = 0.635$ mm and dielectric constant $\varepsilon_r = 9.4$ is used.

The input variables of the model $\boldsymbol{x} = [W_1\ W_2\ L_0\ L_1\ L_2]^T$ are chosen based on the sensitivity data. The model output is the magnitude of S_{21}. Data generation is performed by

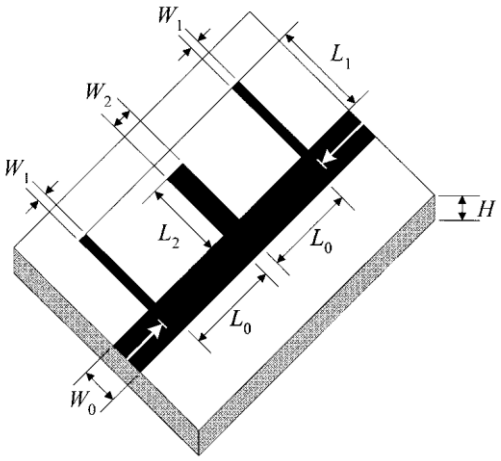

Fig. 9. Bandstop microstrip filter example: EM simulation for data generation.

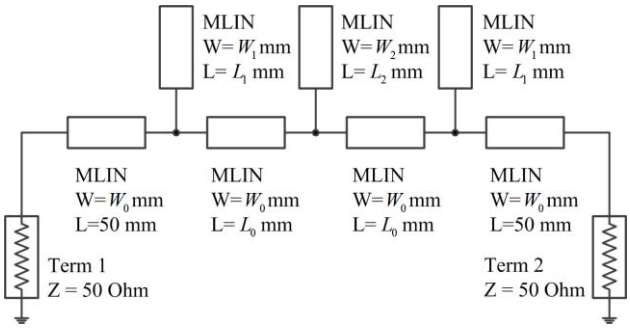

Fig. 10. Bandstop microstrip filter example: the empirical model.

TABLE X
TRAINING DATA AND TESTING DATA FOR PARAMETRIC MODELING OF THE BANDSTOP FILTER EXAMPLE

Input	Training Data		Testing Data	
	Min	Max	Min	Max
	Case 1			
W_1 (mm)	0.15	0.19	0.155	0.185
W_2 (mm)	0.23	0.27	0.232	0.268
L_0 (mm)	2.54	3.35	2.59	3.3
L_1 (mm)	2.54	3.35	2.59	3.3
L_2 (mm)	2.54	3.35	2.59	3.3
Freq (GHz)	5	15	5	15
	Case 2			
W_1 (mm)	0.075	0.23	0.08	0.225
W_2 (mm)	0.11	0.34	0.12	0.33
L_0 (mm)	2.0	3.35	2.05	3.3
L_1 (mm)	2.0	3.35	2.05	3.3
L_2 (mm)	2.0	3.35	2.05	3.3
Freq (GHz)	5	15	5	15

CST Microwave Studio EM simulator using DoE sampling method [30]. The empirical model is the equivalent circuit for the bandstop filter using simple transmission lines, which

TABLE XI

COMPARISON OF THE MODELING RESULTS FOR BANDSTOP FILTER WITH TWO DIFFERENT MODELING RANGES

	Case 1	Case 2
Mapping Structure of the Final Model	Linear input mapping	Nonlinear input mapping + Linear frequency mapping + Nonlinear output mapping
No. of Training Data	81*101	81*101
No. of Testing Data	64*101	64*101
Training Error	2.0%	2.07%
Testing Error	**1.92%**	**1.8%**

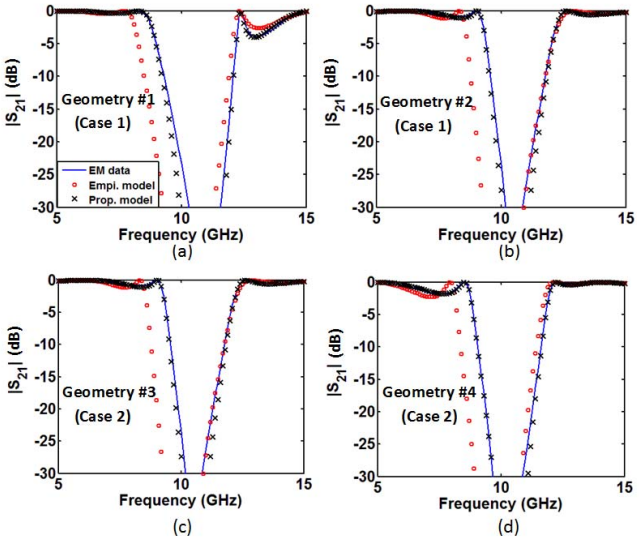

Fig. 11. Bandstop microstrip filter example: modeling results at four different geometrical values (a) $\boldsymbol{x} = [0.185, 0.26, 2.69, 2.59, 2.79]^T$, (b) $\boldsymbol{x} = [0.19, 0.27, 2.59, 2.9, 3.2]^T$, (c) $\boldsymbol{x} = [0.11, 0.17, 2.29, 2.79, 2.79]^T$, and (d) $\boldsymbol{x} = [0.17, 0.14, 2.92, 2.92, 2.67]^T$. The solid line, "o", and "x" in the figures represent the EM simulation data, the empirical model response, and the proposed model response, respectively.

is shown in Fig. 10 [17]. We build our empirical model using formulas based on *Keysight ADS* [36]. The testing error threshold for this bandstop model is set as 2%. After the optimization of the empirical model, the error is still beyond our threshold. Therefore, we proceed to add mappings using our proposed method. The values of $\lambda^{(1)}_{\text{map1},ik}$, $\lambda^{(1)}_{\text{map2},1n}$, and $\lambda^{(1)}_{\text{map3},rq}$ in (10) are set to be 10. Using l_1 optimization, the proposed method can force the weights of the unnecessary mappings to zeros. Therefore, we can remove the unnecessary mappings and produce the most compact mapping structure for the final knowledge-based model to satisfy the accuracy requirement.

For comparison, we develop knowledge-based models for two different sets of geometrical parameter ranges using l_1 optimization, as shown in Table X. For Case 1, the range of the geometrical parameters is small, while the range of the geometrical parameters for Case 2 is large. Table XI and Fig. 11 show the comparison of the modeling results for bandstop filter with two different modeling ranges.

In Case 1, after l_1 optimization, the nonlinear mapping weights $\boldsymbol{v}_{\text{map1}}, \boldsymbol{v}_{\text{map2}}$, and $\boldsymbol{v}_{\text{map3}}$ in three mappings of the knowledge-based model are almost zeros, therefore all the three mappings of the final model are linear mapping. The values of $\lambda^{(2)}_{\text{map1},ij}$, $\lambda^{(2)}_{\text{map2},10}$, $\lambda^{(2)}_{\text{map2},11}$, and $\lambda^{(2)}_{\text{map3},rp}$ in (11) are equal to 10. Besides, the frequency mapping and the output mapping are unit mapping. The proposed algorithm with l_1 optimization uses only a linear input mapping to obtain a knowledge-based model that meets the required accuracy. There is no need for frequency mapping and output mapping in the final model. This modeling result and the observation that a linear mapping is sufficient for a modeling problem with small modeling range are consistent.

For a large modeling range (Case 2), the proposed algorithm with l_1 optimization develops a knowledge-based model with a nonlinear input mapping, a linear frequency mapping, and

TABLE XII

COMPARISON BETWEEN THE MODEL FROM THE PROPOSED AUTOMATED MODELING METHOD AND THE MODELS FROM EXISTING BRUTE FORCE MODELING ALGORITHMS WITH DIFFERENT COMBINATIONS OF MAPPINGS FOR THE BANDSTOP FILTER MODELING PROBLEM

Model Structure	Testing Error	
	Case 1	Case 2
Optimal Model Structure by Proposed Method : **Empirical Model + Linear Input Mapping**	**1.92%**	/
Optimal Model Structure by Proposed Method : **Empirical Model + Nonlinear Input Mapping + Linear Frequency Mapping + Nonlinear Output Mapping**	/	**1.8%**
Empirical Model Alone	21.51%	20.31%
Empirical Model + Linear Input Mapping	1.92%	3.62%
Empirical Model + Nonlinear Input Mapping	1.97%	2.31%
Empirical Model + Linear Output Mapping	17.2%	18.13%
Empirical Model + Nonlinear Output Mapping	17.0%	16.87%
Empirical Model + Linear Input Mapping + Linear Frequency Mapping + Linear Output Mapping	1.79%	3.3%
Empirical Model + Nonlinear Input Mapping + Nolinear Frequency Mapping + Nonlinear Output Mapping	1.39%	1.7%

TABLE XIII
COMPARISON OF THE MODELING TIME FOR BANDSTOP FILTER USING THE EXISTING METHOD IN [24] AND THE PROPOSED METHOD

	Case 1	Case 2
Existing Method in [24]	4.4 h	4.13 h
Proposed Method	2.43 h	3.3 h

a nonlinear output mapping because after l_1 optimization, the nonlinear weights $\boldsymbol{v}_{\text{map1}}$ and $\boldsymbol{v}_{\text{map3}}$ of the input mapping and output mapping are nonzeros while $\boldsymbol{v}_{\text{map2}}$ in the frequency mapping are all zeros. The values of $\lambda^{(2)}_{\text{map1},ij}$ and $\lambda^{(2)}_{\text{map3},rp}$ in (11) are equal to 0, while the values of $\lambda^{(2)}_{\text{map2},10}$ and $\lambda^{(2)}_{\text{map2},11}$ in (11) are equal to 10. For Case 2, we have also generated another set of testing data, which is beyond the training range to test the proposed model, i.e., W_1 : 0.06–0.24 mm, W_2 : 0.1–0.35 mm, L_0 : 1.9–3.43 mm, L_1 : 1.9–3.43 mm, and 1.9–3.43 mm. The testing error is 4.53%. Within the training range, the proposed model has small testing error, which satisfies the user-desired accuracy. Beyond the training range, the proposed model still has reasonable accuracy, which means the extrapolation capability of the model is retained because of the embedded knowledge.

Table XII shows the comparison between the proposed automated modeling method and the existing brute force modeling method with different combinations of input, frequency, and output mapping neural networks. In reality, the number of the possible combinations of the mappings in a knowledge-based model is more than that listed in Table XII. We also compare the modeling time between the existing method in [24] and our proposed method, as shown in Table XIII. The modeling results demonstrate that for different modeling examples, our proposed method with l_1 optimization can develop knowledge-based models with different combinations of mappings, which makes the model development process very flexible. The proposed algorithm increases the modeling efficiency and develops a knowledge-based model with the compact mapping structure in a shorter time than existing knowledge-based modeling algorithms.

V. CONCLUSION

In this paper, a unified automated parametric modeling algorithm using combined knowledge-based neural network and l_1 optimization has been proposed. In our method, we propose a new unified knowledge-based model structure and new sensitivity formulas. The proposed knowledge-based model combines the learning ability of the neural network with the convenience and simplicity of the empirical model. We use the properties of l_1 norm to propose a training method that can automatically determine whether a mapping is necessary, or whether a mapping is linear or nonlinear. Instead of sequential trials of linear and nonlinear mappings every time for every mapping structure as in the existing literature, the proposed algorithm uses l_1 optimization to automatically determine the final model structure. Compared to existing knowledge-based modeling methods, the proposed algorithm increases the knowledge-based modeling efficiency and speeds up the model development process.

REFERENCES

[1] H. Kabir, L. Zhang, M. Yu, P. H. Aaen, J. Wood, and Q.-J. Zhang, "Smart modeling of microwave devices," *IEEE Microw. Mag.*, vol. 11, no. 3, pp. 105–118, May 2010.

[2] J. E. Rayas-Sanchez, "EM-based optimization of microwave circuits using artificial neural networks: The state-of-the-art," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 1, pp. 420–435, Jan. 2004.

[3] Q. J. Zhang and K. C. Gupta, *Neural Networks for RF and Microwave Design*. Norwood, MA, USA: Artech House, 2000.

[4] W. Liu, W. Na, L. Zhu, and Q.-J. Zhang, "A review of neural network based techniques for nonlinear microwave device modeling," in *Proc. IEEE MTT-S Int. Conf. Numer. Electromagn. Multiphys. Modeling Optim. (NEMO)*, Beijing, China, Jul. 2016, pp. 1–2.

[5] S. K. Mandal, S. Sural, and A. Patra, "ANN- and PSO-based synthesis of on-chip spiral inductors for RF ICs," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 27, no. 1, pp. 188–192, Jan. 2008.

[6] T. Liu, S. Boumaiza, and F. M. Ghannouchi, "Dynamic behavioral modeling of 3G power amplifiers using real-valued time-delay neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 3, pp. 1025–1033, Mar. 2004.

[7] M. Isaksson, D. Wisell, and D. Ronnow, "Wide-band dynamic modeling of power amplifiers using radial-basis function neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 53, no. 11, pp. 3422–3428, Nov. 2005.

[8] J. Wood, D. E. Root, and N. B. Tufillaro, "A behavioral modeling approach to nonlinear model-order reduction for RF/microwave ICs and systems," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 9, pp. 2274–2284, Sep. 2004.

[9] S. A. Sadrossadat, Y. Cao, and Q.-J. Zhang, "Parametric modeling of microwave passive components using sensitivity-analysis-based adjoint neural-network technique," *IEEE Trans. Microw. Theory Techn.*, vol. 61, no. 5, pp. 1733–1747, May 2013.

[10] F. Feng, C. Zhang, J. G. Ma, and Q.-J. Zhang, "Parametric modeling of EM behavior of microwave components using combined neural networks and pole-residue-based transfer functions," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 1, pp. 60–77, Jan. 2016.

[11] Y. Cao, S. Reitzinger, and Q.-J. Zhang, "Simple and efficient high-dimensional parametric modeling for microwave cavity filters using modular neural network," *IEEE Microw. Wireless Compon. Lett.*, vol. 21, no. 5, pp. 258–260, May 2011.

[12] F. Wang and Q.-J. Zhang, "Knowledge-based neural models for microwave design," *IEEE Trans. Microw. Theory Techn.*, vol. 45, no. 12, pp. 2333–2343, Dec. 1997.

[13] J. W. Bandler, M. A. Ismail, J. E. Rayas-Sánchez, and Q.-J. Zhang, "Neuromodeling of microwave circuits exploiting space-mapping technology," *IEEE Trans. Microw. Theory Techn.*, vol. 47, no. 12, pp. 2417–2427, Dec. 1999.

[14] J. W. Bandler *et al.*, "Space mapping: The state of the art," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 1, pp. 337–361, Jan. 2004.

[15] S. Koziel, Q. S. Cheng, and J. W. Bandler, "Space mapping," *IEEE Microw. Mag.*, vol. 9, no. 6, pp. 105–122, Dec. 2008.

[16] M. H. Bakr, J. W. Bandler, M. A. Ismail, J. E. Rayas-Sanchez, and Q.-J. Zhang, "Neural space-mapping optimization for EM-based design," *IEEE Trans. Microw. Theory Techn.*, vol. 48, no. 12, pp. 2307–2315, Dec. 2000.

[17] S. Koziel, J. W. Bandler, and K. Madsen, "Space mapping with adaptive response correction for microwave design optimization," *IEEE Trans. Microw. Theory Techn.*, vol. 57, no. 2, pp. 478–486, Feb. 2009.

[18] V. Gutiérrez-Ayala and J. E. Rayas-Sánchez, "Neural input space mapping optimization based on nonlinear two-layer perceptrons with optimized nonlinearity," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 20, no. 5, pp. 512–526, Sep. 2010.

[19] J. E. Rayas-Sánchez and V. Gutiérrez-Ayala, "EM-based Monte Carlo analysis and yield prediction of microwave circuits using linear-input neural-output space mapping," *IEEE Trans. Microw. Theory Techn.*, vol. 54, no. 12, pp. 4528–4537, Dec. 2006.

[20] S. Koziel, J. W. Bandler, and K. Madsen, "A space-mapping framework for engineering optimization—Theory and implementation," *IEEE Trans. Microw. Theory Techn.*, vol. 54, no. 10, pp. 3721–3730, Oct. 2006.

[21] L. Zhang, J. Xu, M. C. E. Yagoub, R. Ding, and Q.-J. Zhang, "Efficient analytical formulation and sensitivity analysis of neuro-space mapping for nonlinear microwave device modeling," *IEEE Trans. Microw. Theory Techn.*, vol. 53, no. 9, pp. 2752–2767, Sep. 2005.
[22] D. Gorissen, L. Zhang, Q.-J. Zhang, and T. Dhaene, "Evolutionary neuro-space mapping technique for modeling of nonlinear microwave devices," *IEEE Trans. Microw. Theory Techn.*, vol. 59, no. 2, pp. 213–229, Feb. 2010.
[23] W. Na and Q.-J. Zhang, "Automated knowledge-based neural network modeling for microwave applications," *IEEE Microw. Wireless Compon. Lett.*, vol. 24, no. 7, pp. 499–501, Jul. 2014.
[24] W. Na and Q. J. Zhang, "Unified automated knowledge-based neural network modeling for microwave devices," in *Proc. IEEE MTT-S Int. Conf. Numer. Electromagn. Multiphys. Modeling Optim. (NEMO)*, Ottawa, ON, Canada, Aug. 2015, pp. 1–3.
[25] V. K. Devabhakruni, M. C. E. Yagoub, and Q.-J. Zhang, "A robust algorithm for automatic development of neural-network models for microwave applications," *IEEE Trans. Microw. Theory Techn.*, vol. 49, no. 12, pp. 2282–2291, Dec. 2001.
[26] J. W. Bandler, S. H. Chen, and S. Daijavad, "Microwave device modeling using efficient l_1 optimization: A novel approach," *IEEE Trans. Microw. Theory Techn.*, vol. 34, no. 12, pp. 1282–1293, Dec. 1986.
[27] J. W. Bandler, W. Kellermann, and K. Madsen, "A nonlinear l_1 optimization algorithm for design, modeling, and diagnosis of networks," *IEEE Trans. Circuits Syst.*, vol. 34, no. 2, pp. 174–181, Feb. 1987.
[28] J. Hald and K. Madsen, "Combined LP and quasi-Newton methods for nonlinear l_1 optimization," *J. Numer. Anal.*, vol. 22, no. 1, pp. 68–80, 1985.
[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362.
[30] S. R. Schmidt and R. G. Launsby, *Understanding Industrial Designed Experiments*. Colorado Springs, CO, USA: Air Force Academy, 1992.
[31] M. L. Strydom and J. P. Eberhard, "Reliability and efficiency in computational electromagnetics: How CEM software meets the future needs of engineers," in *Proc. IEEE Antennas Propag. Soc. Int. Symp.*, Honolulu, HI, USA, Jun. 2007, pp. 5451–5454.
[32] A. Vasylchenko, Y. Schols, W. D. Raedt, and G. A. E. Vandenbosch, "Quality assessment of computational techniques and software tools for planar-antenna analysis," *IEEE Antennas Propag. Mag.*, vol. 51, no. 1, pp. 23–38, Feb. 2009.
[33] W. H. Tu and K. Chang, "Microstrip elliptic-function low-pass filters using distributed elements or slotted ground structure," *IEEE Trans. Microw. Theory Techn.*, vol. 54, no. 10, pp. 3786–3792, Oct. 2006.
[34] R. Esfandiari, D. Maku, and M. Siracusa, "Design of interdigitated capacitors and their application to gallium arsenide monolithic filters," *IEEE Trans. Microw. Theory Techn.*, vol. 31, no. 1, pp. 57–64, Jan. 1983.
[35] F. Feng, C. Zhang, V.-M.-R. Gongal-Reddy, Q.-J. Zhang, and J. Ma, "Parallel space-mapping approach to EM optimization," *IEEE Trans. Microw. Theory Techn.*, vol. 62, no. 5, pp. 1135–1148, May 2014.
[36] E. Hammerstad and O. Jensen, "Accurate models for microstrip computer-aided design," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Washington, DC, USA, May 1980, pp. 407–409.
[37] M. Kirschning and R. H. Jansen, "Accurate wide-range design equations for the frequency-dependent characteristic of parallel coupled microstrip lines," *IEEE Trans. Microw. Theory Techn.*, vol. 32, no. 1, pp. 83–90, Jan. 1984.

Weicong Na (S'13) was born in Qiqihaer, Heilongjiang, China, in 1989. She received the B.Eng. degree from Tianjin University, Tianjin, China, in 2012. She is currently pursuing the Ph.D. degree at the School of Electronic Information Engineering, Tianjin University, and the cotutelle Ph.D. program at the Department of Electronics, Carleton University, Ottawa, ON, Canada.

Her current research interests include microwave circuit design and modeling, automated neural network model generation algorithms, and electromagnetic field knowledge-based modeling and optimization.

Feng Feng (S'13) was born in Huludao, Liaoning, China, in 1990. He received the B.Eng. degree from Tianjin University, Tianjin, China, in 2012. He is currently pursuing the Ph.D. degree at the School of Electronic Information Engineering, Tianjin University, and the cotutelle Ph.D. program at the Department of Electronics, Carleton University, Ottawa, ON, Canada.

His current research interests include microwave circuit design and modeling, optimization theory and algorithms, space mapping and surrogate model optimization, and EM field simulation and optimization.

Chao Zhang (S'14) was born in Xinyang, Henan, China, in 1990. He received the B.Eng. and M.Eng. degrees from Tianjin University, Tianjin, China, in 2012 and 2014, respectively. He is currently pursuing the Ph.D. degree at the Department of Electronics, Carleton University, Ottawa, ON, Canada.

In 2010, he was an Exchange Student with the National Taipei University of Technology, Taipei, Taiwan. His current research interests include design and optimization of microwave circuits, space mapping, and cognition-driven space mapping.

Mr. Zhang was a recipient of the National Scholarship for Graduate Students in China in 2013.

Qi-Jun Zhang (S'84–M'87–SM'95–F'06) received the B.Eng. degree from the Nanjing University of Science and Technology, Nanjing, China, in 1982, and the Ph.D. degree in electrical engineering from McMaster University, Hamilton, ON, Canada, in 1987.

From 1982 to 1983, he was with the System Engineering Institute, Tianjin University, Tianjin, China. From 1988 to 1990, he was with the Optimization Systems Associates Inc., Dundas, ON, Canada. In 1990, he joined the Department of Electronics, Carleton University, Ottawa, ON, Canada, where he is currently a Full Professor. On leave from Carleton University, he has been with the School of Electronic Information Engineering, Tianjin University.

Dr. Zhang is a member on the Editorial Boards of the IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES and also a member of the Technical Committee on CAD (MTT-1) of the IEEE Microwave Theory and Techniques Society. He is a Fellow of the Electromagnetics Academy and the Canadian Academy of Engineering.