A knowledge-based neuromodeling using space mapping technique: Compound space mapping-based neuromodeling

Murat Simsek* † and N. Serap Sengor

Istanbul Technical University, Faculty of Electrical and Electronic Engineering, Electronics Engineering Department, Maslak, TR-34469, Istanbul, Turkey

SUMMARY

This paper presents two new methods, space mapping (SM) with prior knowledge input (PKI-D) with difference and compound space mapping-based neuromodeling. Both methods combine two powerful techniques, space mapping-based neuromodeling and PKI-D with difference. The knowledge-based modeling methods in the RF/microwave literature merge the prior knowledge about the device to be modeled with neural network structures while a knowledge-based method, SP, focuses on reducing the computational burden. The main advantage of the proposed methods over these already existing knowledge-based methods are their better extrapolation capability and reduced number of training set data. The simulation results obtained reveal that both methods decrease the cost of training and improve the extrapolation capability and output performance of the SP-based neuromodeling. Copyright © 2007 John Wiley & Sons, Ltd.

Received 6 March 2007; Revised 3 July 2007; Accepted 9 August 2007

KEY WORDS: knowledge-based neuromodeling; artificial neural networks; space mapping

1. INTRODUCTION

New techniques and methods that would fulfill the need of computationally fast and accurate models have been sought in modeling especially radio frequency (RF) devices, as market needs more and more usage of these devices. To provide such techniques and methods, approach in modeling these devices have gone through reconsideration during the past decade. Now, the time spent during design process is as important as the efficiency of the models, so especially artificial neural network (ANN)-based structures, which are capable of nonlinear modeling and fast once the training phase is completed, have begun to be used effectively even in toolboxes of design-oriented simulators [1–3]. Nevertheless, there is a need for the improvement of these
models as the computational demand for training set is still a problem to solve. Space mapping (SM) technique as it is proposed in [4] aims at resolving this problem in general. Another aspect that needs to be improved is ANN-based models with better extrapolation capability. As ANN models are black-box models, even though their interpolation capability is better than most conventional modeling approach as look-up tables, polynomial approximations, they are highly dependent on training set. A way to overcome this flaw is to make use of the already existing knowledge about the device to be modeled along with ANN-based modeling. Thus, different knowledge-based methods as prior knowledge input (PKI), source difference (SD), prior knowledge input with difference (PKI-D) and equivalent circuit-state-space equation-neural network (EC-SSE-NN) have been proposed [5–8].

In this work, two new methods that focus on decreasing the training set burden and improving the extrapolation capability are proposed. In these models, to reduce the training set burden, SM is used, whereas to implement the prior knowledge about the device to be modeled, a knowledge-based method PKI-D is integrated. The proposed methods are hybrid methods-based on PKI-D and SM-based neuromodeling (SMN), namely SMN with PKI-D method and compound space mapping-based neuromodeling (CSMN). Both models combine the advantages of SMN and PKI-D and improve the two aspects mentioned above. Thus, first PKI-D and SMN will be introduced in Section 2, and then in Section 3, two proposed methods SMN with PKI-D method and CSMN will be introduced. Both methods are inspired by the idea of combining the advantages of SMN and PKI-D in one structure. While SMN with PKI-D does this in discrete processing steps, CSMN does it in an integrated manner. In order to show that these newly proposed methods improve the extrapolation capability while decreasing the computational burden of forming the training set, two different modeling applications are introduced and simulation results for these applications are given. The simulation results will especially exploit the advantage of the proposed methods over SMN when extrapolation capability is considered. Another important result will be on the number of training data needed to get the models. Further comparison of the proposed methods and the other methods are also given in [7, 9]. While one of the applications deals with modeling the characteristic impedance of micro-strip line, the other is on function approximation. Modeling the Branin function is considered to show that even though the starting point is to improve modeling of RF devices, the methods proposed can be used in different applications where the problem can be expressed as a function approximation problem.

2. A PRELIMINARY ON KNOWLEDGE-BASED METHODS

There are various approaches in modeling RF/microwave devices. While from ElectroMagnetic (EM) simulators highly accurate models can be obtained, from equivalent circuits and empiric relations computationally efficient models can be derived. Since EM simulators give almost exact solutions they are named ‘fine models’ while other approaches as equivalent circuits, empiric relations are named ‘coarse models’. The best approach should be capable of deriving a model as accurate as ‘fine model’ and as computationally efficient as ‘coarse model’.

ANN structures for modeling RF/microwave devices have numerous advantages over traditional modeling approaches [1]. Their main advantage is once trained with a well-selected, proper training data obtained from EM simulators, they give, in the training data interval, as good results as the data set even for the parameter values that were not in the training set. This is
due to their high function approximation property and generalization capability. Once the training phase is completed ANN models are computationally very efficient as they involve very simple computations compared with EM simulators. With these properties, ANN structures are good candidates for model deriving, but they also have disadvantages. As ANN models form the input/output relation just by using training data through the training phase no knowledge about the phenomena to be modeled is used. Hence, even though they give satisfactory results for interpolation, they are poor in extrapolation. Another drawback is its accuracy is very much dependent on the training data, so there is need of large training data with well-selected values over a large interval.

In order to overcome these disadvantages of ANN structures and to benefit from the advantages they offer, various knowledge-based methods that make use of ‘coarse model’ are proposed [1, 4–9]. Amongst these, SM technique introduces a procedure to form a mapping which when used with ‘coarse model’ gives accurate results while using ‘fine model’ efficiently. While SM can be employed especially to form efficient training set for ANN structures, other knowledge-based methods as PKI, SD and PKI-D are effective for implementing the already existing relations and associations about the device to be modeled. In the sequence, first PKI-D method will be introduced and then SMN will be reviewed to ease the idea behind the proposed methods given in Section 3.

2.1. Prior knowledge input with difference method

PKI-D is a hybrid method combining the advantages of two methods, PKI and SD [7]. It establishes input/output relationship not only considering model inputs as inputs but also through extra input obtained from ‘coarse model’ output as shown in Figure 1(a). The ANN structure in the method is trained to comprehend the difference between the ‘fine model’ and the ‘coarse model’. ‘Fine model’ gives target response while ‘coarse model’ exploits approximate output compared to target response.

In PKI-D method like SD, the final output to be utilized is obtained by summing PKI-D output and ‘coarse model’ output which can be followed from Figure 1(b). While extra input reduces model complexity and adds regional input/output relationship to PKI-D, the learning range is narrowed since the difference between ‘fine model’ output and ‘coarse model’ output is

Figure 1. PKI-D model structure: (a) training phase of PKI-D and (b) final model of PKI-D.
used as the desired output; thus, the learning process is improved. Hence, this hybrid method combines the advantages of both PKI and SD methods. In [7], the comparison of PKI-D with plain ANN model, SD and PKI methods are given and the results obtained there show that PKI-D is superior to the others.

2.2. Space mapping-based neuromodeling

The aim of SMN is to find an appropriate mapping from input space of the ‘fine model’ to input space of the ‘coarse model’, and ANN structure establishes this mapping which is denoted by \( P(.) \). Once this mapping is determined the expectation is the ‘coarse model’ will generate results as adequate as the ‘fine model’ without much computational burden. The vectors \( x_c \) and \( x_f \) represent the input parameters of the ‘coarse model’ and ‘fine model’, respectively. \( R_c(x_c) \) represents the corresponding ‘coarse model’ response \( R_c \) and \( R_f(x_f) \) represents the corresponding ‘fine model’ response \( R_f \). Initial base points for ‘fine model’ input space are obtained from star distribution [1] in which optimal ‘coarse model’ point \( x_c^* \) is used as the mid-point value. All steps of the SMN method have been stated in [1]. For error evaluation, i.e. to evaluate the difference between fine model output and coarse model output, the same upper bound is used in finding the new \( x_f \) and in deciding to stop the training phase of ANN structure that establishes the mapping between two model input spaces. In addition, the error upper bound used to determine ANN structure has been decreased at each iteration step in order to provide SMN to converge to optimal response of the ‘coarse model’. In general, feed-forward neural network structures as multilayer perceptron (MLP) or radial basis function (RBF) networks are used to set up the mapping \( P(.) \) between ‘fine model’ and ‘coarse model’ input spaces. The formulation of feed-forward ANN structure is as follows:

\[
y_i = \sum_{j=1}^{n_h} a_j \psi_j(w_j^T x_i + \theta_j)
\]

where \( n_h \) is the number of hidden neurons and \( \Psi_j(\cdot) \) is the activation function, \( w_j \) weight vector and \( \theta_j \) the bias associated with the \( j \)th hidden layer neuron and \( a_j \)'s correspond to output layer weights. The training phase to determine the weights and biases can be formulated as an optimization problem as follows:

\[
[W, \theta, A] = \arg\min_{w,\theta,A} \{ ||e_1^T e_2^T \cdots e_N^T || \}
\]

where

\[
e_i = R_c(x_f) - R_f(x_f^{(i)})
\]

Once the optimization problem stated by (2) is solved, the parameters \( W = [w_1 \ w_2 \ \cdots \ w_{n_h}] \), \( \theta = [\theta_1 \ \theta_2 \ \cdots \ \theta_m] \), \( A = [a_1 \ a_2 \ \cdots \ a_n] \) necessary to set up the mapping \( P(.) \) are determined.

Thus, the mapping \( P(.) \) is constructed by ANN structure and once this phase is complete, \( x_c \) can be determined by \( P(.) \) using \( x_f \). The SMN response \( Y_c \) is obtained using ‘coarse model’ response function \( R_c \) while the input of ‘coarse model’ \( x_c \) is determined from \( x_f \) using the mapping \( P(.) \). This model is constructed with less ‘fine model’ response than conventional neuromodel, since instead of determining an ANN structure through training which requires a large ‘fine model’ response only mapping \( P(.) \) is formed and used with ‘coarse model’. While determining \( P(.) \) through training phase, ‘coarse model’ is not adapted and this decreases cost of coarse approximation, since it is used only to form input/output relation. The steps of the SMN
method can be followed from block on the right-hand side of Figure 4 which corresponds to the processes of SMN.

3. PROPOSED METHODS

Each of the knowledge-based methods summarized in Section 2 improve one of the two main drawbacks of ANN-based modeling, but do not resolve both at the same time. While PKI-D improves the extrapolation capability [4], SMN decreases the computational burden of utilizing 'fine model'. The main purpose of this section is to introduce two new methods capable of improving both. While the first one uses both methods consecutively, the other combines the advantages of PKI-D and SMN in one structure where the training phases of ANN structures are carried out in the same computation loop. As mentioned above by these proposed methods, the need for fine model responses is decreased as the characterization of SM fine model responses are used only when needed. This advantage of the proposed methods can be followed from [9], where a comparison between different methods is given for the characteristic impedance modeling of micro-strip line.

3.1. SMN with PKI-D method

SMN method exploits less training data and has less computational cost than other modeling approaches. The data that have been used in SMN to obtain a better model performance can be used in PKI-D method as a training set. For these reasons, these two different methods will be processed in sequential time steps. The new hybrid learning model is called ‘SMN with PKI-D’. In this method, PKI-D exploits less number of but more relevant training data to acquire fine model behavior, and some important aspects such as generalization, extrapolation capability and model accuracy also are improved compared to the other knowledge-based methods [9].

First, SMN part is processed and a mapping \( P(\cdot) \) is formed. Then, initial points and new extracted points are combined in a set which is used during the training phase of PKI-D. As SMN extracts relevant input points for PKI-D through fine model, the accuracy of PKI-D as a model is improved compared to PKI-D model without beneficiating SMN. This approach in the training phase of PKI-D solves the problems that are related to finding appropriate training set and reducing the number of training data. Inputs and outputs for SMN with PKI-D and training phases are given in Figure 2. PKI-D acquires the difference \( (D) \) between fine and coarse model responses while input of both fine and coarse models is the input space of original problem. \( Y_c \), which is applied as an input to PKI-D, helps to set relationships between \( D \) and \( x_f \) and also reduces the problem complexity during the PKI-D training phase. The final model obtained as the result of SMN with PKI-D approach is given in Figure 3. SMN and PKI-D run sequentially in the final model where model input is \( x_f \) and model output is \( Y_d \). The output of the model obtained at the end of the training phases in SMN with PKI-D method will be as effective as \( D \).

All the steps of the algorithm for SMN with PKI-D are shown explicitly in Figure 4. In this figure, there are two parts; in one of them SMN process is demonstrated while in the other the training of PKI-D is illustrated. After the process related to SMN is completed and the mapping \( P(\cdot) \) is formed, the process related to PKI-D starts. During the training phase of PKI-D, inputs determined as the result of SMN are used.
3.2. Compound space mapping-based neuromodeling method

Even though SMN with PKI-D method combines the advantages of SMN with a knowledge-based structure, two different methods are used discretely. In order to combine both in one structure CSMN is proposed. Here, PKI-D is chosen as the knowledge-based method since its extrapolation and interpolation ability is superior to other knowledge-based methods such as PKI and SD [7]. As stated in Section 3.1, input space of the PKI-D consists of $x_f$ and $Y_c$ and its output space consists of $D$. In Figure 5, block diagram for training phase of CSMN is given and in Figure 6 all the steps of the algorithm are shown. Comparing Figures 4 and 6, the difference between two proposed models can be followed. While in Figure 4, PKI-D and SMN are two completely different processes, in Figure 6, these two are combined in one step. Training part of CSMN has been accomplished in three steps. First, SMN step, which creates appropriate mapping $P(\cdot)$ from $x_f$ to $x_c$, is completed. In the second step, PKI-D modeling is formed, which determines $D$ using $x_f$ and $Y_c$. The last step is finding new $x_f$ using $P(\cdot)$ and $R_c$ blocks of Figure 4. The aim of the last step is to find an appropriate $x_f$ which satisfies a priori set-up error upper bound. The final model of CSMN would be just like the one in Figure 3, only difference would be that instead of $Y_d$ the name of the output would be $Y_{dc}$. Model input parameter is $x_f$ and

Figure 2. Block diagram of training process for SMN with PKI-D. The first block on the left-hand side corresponds to the training phase where mapping $P(\cdot)$ is formed, and the second block corresponds to the training phase of PKI-D.

Figure 3. Block diagram of the final model for SMN with PKI-D. Once the training to determine $P(\cdot)$ and the ANN structure within PKI-D is completed, the final model is used.
Figure 4. SMN with PKI-D flowchart. The inner block in the block on the left-hand side corresponds to training of ANN structure within PKI-D and the outer block corresponds to forming the SMN with PKI-D model. The block on the right-hand side corresponds to SMN forming the mapping $P(\cdot)$.

Figure 5. Block diagram of training process for CSMN. The blocks $R_f$, $R_c$ correspond to 'fine model' and 'coarse model', respectively. The blocks $P$ and PKI-D correspond to the mapping to be formed by SMN technique and the PKI-D structure implemented in SMN, respectively.
model output is $Y_{dc}$, which will be used as equivalent of ‘fine model’ output $Y_f$ once the CSMN model is formed.

As in SMN, the same upper bound is used both to find new $x_f$ and to decide on stopping the training phase of ANN structure that establishes the mapping $P(\cdot)$ between two model input spaces. Furthermore, maximum error upper bound has been reduced at each iteration step of SMN and PKI-D modeling. Therefore, error upper bound has been improved step by step during training. After training, $Y_e$ and $Y_D$ have been used to get desired output $Y_{dc}$. All results of the CSM have been investigated using the model output $Y_{dc}$.

4. APPLICATIONS

In order to show that the expected improvements as decreasing the number of data needed for training and obtaining a better extrapolation capability are fulfilled, simulations are carried out for two different applications. In both applications the same implementations of SMN with
PKI-D method and CSMN method are used. In both methods, while forming the mapping $P()$ through SMN, MLP structure with two hidden layers is used [10]. The numbers of neurons in the first layer and second layer are 30 and 20, respectively, the activation function is logistic function, the learning rate for two hidden layers and output layer is 0.07, 0.08 and 0.09, respectively. The training of MLP is accomplished by using error back-propagation with momentum term where the learning rate related to momentum term is 0.1. For the ANN structure in PKI-D, RBF is preferred as it has better adaptation property to sudden changes in the training set. Gaussian function is used as the activation function and the number of hidden neurons for RBF structure is 50. The training is carried out for all parameters of the RBF; these are weights, positions and the spread of centers and all of them are adapted through learning. The learning rule is similar to error back-propagation but this time the parameters of Gaussian function along with weights are updated [10]. The results are given only for the test sets that are formed using different data selected inside the training data interval for interpolation and using data selected outside the training data set interval for extrapolation. For both methods the codes are formed as m-files in MATLAB®.

### 4.1. Modeling characteristic impedance of micro-strip line

The first application is to determine the variation of the characteristic impedance of thin film micro-strip line due to changes in parameters: conductor thickness $(t)$, conductor width $(w)$ and substrate height $(h)$. Hence, the input data will be of a three-dimensional vector and the output dimension is one. The ‘fine model’ response corresponding to the characteristic impedance of thin film micro-strip line is obtained using TraceSim and the ‘coarse model’ response is obtained using the empirical formula given in [15]. While the interpolation interval is $[10^{-3}, 12.9 \times 10^{-3}]$, $[0.4 \times 10^{-3}, 1.59 \times 10^{-3}]$ and $[4.4 \times 10^{-3}, 9.8 \times 10^{-3}]$ (m) for $t$, $w$ and $h$, respectively, the extrapolation intervals are $[(0.9 \times 10^{-3}, 10^{-3}], [12.9 \times 10^{-3}, 14 \times 10^{-3}]]$, $[0.2 \times 10^{-3}, 0.4 \times 10^{-3}]$, $[1.59 \times 10^{-3}, 1.7 \times 10^{-3}]$ and $[3 \times 10^{-3}, 4.4 \times 10^{-3}], [9.8 \times 10^{-3}, 10.7 \times 10^{-3}]$ for $t$, $w$ and $h$, respectively. Figure 7 shows that only nine extra ‘fine model’ values in addition to the initial base point are needed to obtain a model using SMN with PKI-D and CSMN methods. This number is very low compared to 100 ‘fine model’ values needed to train neural network structure in PKI-D method [9]. In Figures 8 and 9 comparison of ‘fine model’, ‘coarse model’, SMN, CSMN and SMN with PKI-D methods are given for test data in the interpolation and extrapolation intervals, respectively. It can be followed from these figures that CSMN and SMN with PKI-D method responses follow the fine model response very closely. The overall improvement for test sets can be also followed from Figure 10, where each sample corresponds to three-dimensional input vectors. All the results are further summarized in Table I.

### 4.2. Function approximation

In order to explore the efficiency of the proposed methods in a general setting, a function approximation problem is considered and the Branin function is used as an example. The graphics of the Branin function is given in Figure 11 and its analytic expression is presented in the following equation:

$$y = \left(x_2 - \frac{5x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$$  \hspace{1cm} (4)
As can be followed from Figures 8(a) and 9(a) the behavior of ‘coarse model’ is very similar to that of ‘fine model’ in the first application; hence, to observe the efficiency of the proposed methods when the ‘coarse model’ is quite different from ‘fine model’, a ‘coarse model’ for the
Branin function is created using the expression given in the following equation and the difference between the two models can be followed from Figure 12:

$$y = 0.95x_2 - \frac{5x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6 + 10 \left( 1 - \frac{1}{8\pi} \right) \left( \cos \frac{x_1}{3} \right)^2 - \left( \sin \frac{x_1}{3} \cos \frac{x_1}{4} + \cos \frac{x_1}{3} \sin \frac{x_1}{4} \right)^2 + 12 \quad (5)$$

Here, the created ‘coarse model’ is complicated than ‘fine model’, but this example is created just to give an idea about the effectiveness of the proposed methods when the ‘coarse model’ and

Figure 9. Comparison of test results for characteristic impedance in the extrapolation interval. The results are given for two fixed values of input parameter $t$ which are also taken in the extrapolation interval while the ratio of two other input parameters is changed. The $y$-axis corresponds to characteristic impedance in $\Omega$ and the $x$-axis corresponds to dimensionless values of the ratio of substrate height $(h)$ and conductor width $(w)$: (a) comparison of $Y_f$ ‘fine model’ response, $Y_c$ SMN method response and $Y_e$ ‘coarse model’ response with input data are same as the ‘fine model’ and (b) comparison of $Y_f$ ‘fine model’ response, $Y_c$ SMN method response and $Y_{dc}$ CSMN method response, $Y_d$ SMN with PKI-D method response.

Figure 10. Error percentage for test set of characteristic impedance, where the difference between $Y_f$ ‘fine model’ and $Y_c$ ‘coarse model’ response, $Y_{dc}$ CSMN method response and $Y_d$ SMN with PKI-D method response are given: (a) extrapolation interval and (b) interpolation interval.
fine model' outputs are not very similar. Figure 13 shows that for this application, very few 'fine model' values are enough, and the comparison of the methods with 'fine model' and 'coarse model' responses is given in Figures 14 and 15, for interpolation and extrapolation intervals, respectively. While the interpolation interval can be followed from Figure 12(b), the extrapolation interval is
\[
\left(\frac{5}{10^{2}}\right) \times (x_{2} - (\frac{5}{10^{2}}) x_{1}) + \left(\frac{5}{10^{2}}\right) x_{2} - (\frac{5}{10^{2}}) x_{1} + 10
\]

The overall improvement the proposed methods provide can be followed from Figure 16 and all results are summarized in Table II.

5. CONCLUSION AND DISCUSSIONS

The simulation results for modeling characteristic impedance of micro-strip line and the Branin function are summarized in Tables I and II, respectively. In both tables, the test results for 'coarse model', SMN, SMN with PKI-D and CSMN are compared. Twenty different trials have been carried out to have statistical meaningful results as each time an ANN structure is trained.
the training phase begins with randomly selected, different weight values; thus at the end of the training phase the ANN parameters obtained are different for each trial. Hence, the test set results given in the tables are the means of 20 different trials. The number of test data is 20 for interpolation and extrapolation for the Branin function approximation, and, respectively,
25 and 10 for characteristic impedance application. As can be followed from the tables, SMN with PKI-D and CSMN methods outperform the other methods, both in interpolation and in extrapolation sets.

Figure 14. Comparison of test results for the Branin function approximation in the interpolation interval. The results are given for two fixed values of parameter $x_2$ while the other parameter $x_1$ is changed. The $y$-axis corresponds to the Branin function value and the $x$-axis corresponds to $x_1$: (a) comparison of $Y_f$ 'fine model' response, $Y_c$ SMN method response and $Y_e$ 'coarse model' response with input data are same as the 'fine model' and (b) comparison of $Y_f$ 'fine model' response, $Y_c$ SMN method response and $Y_{dc}$ CSMN method response, $Y_d$ SMN with PKI-D method response.

Figure 15. Comparison of test results for the Branin function approximation in the extrapolation interval. The results are given for two fixed values of parameter $x_2$ which are also taken in the extrapolation interval while the other parameter $x_1$ is changed. The $y$-axis corresponds to the Branin function value and the $x$-axis corresponds to $x_1$: (a) comparison of $Y_f$ 'fine model' response, $Y_c$ SMN method response and $Y_e$ 'coarse model' response with input data are same as the 'fine model' and (b) comparison of $Y_f$ 'fine model' response, $Y_c$ SMN method response and $Y_{dc}$ CSMN method response, $Y_d$ SMN with PKI-D method response.
The Branin function approximation application not only implies that these methods can be used for applications other than RF/microwave modeling but also reveals that the proposed methods give satisfactory results even when ‘coarse model’ responses are not much similar with ‘fine model’ responses. This problem of dissimilarity between ‘coarse model’ and ‘fine model’ responses has been considered for SM technique and aggressive and hybrid aggressive methods has been proposed to overcome inconveniences in such cases [11, 12]. SMN with PKI-D and CMSN methods overcome this problem just using SM technique.

The aim of this work is to improve the ANN-based modeling especially considering two aspects: decreasing the number of data in a training set and improving the extrapolation ability. Even though there are numerous works on SM, most of these deal with improving the SM technique [1, 11–14]. In this work, the emphasis is on improving ANN-based modeling especially employing the knowledge-based structures along with SMN. The results obtained reveals that using hybrid methods improves the drawbacks of ANN-based modeling. Of course, these improvements have pay offs and, the complexity of the model structure and the complexity

![Figure 16. Error percentage for test set of the Branin function, where the difference between \( Y_f \) ‘fine model’ and \( Y_c \) ‘coarse model’ response, \( Y_{dc} \) CSMN method response and \( Y_d \) SMN with PKI-D method response: (a) extrapolation interval and (b) interpolation interval.](image)

Table II. Test results for the Branin function approximation.

<table>
<thead>
<tr>
<th></th>
<th>Coarse model</th>
<th>SMN</th>
<th>SMN with PKI-D</th>
<th>CSMN</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Interpolation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sample = 20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max error</td>
<td>0.19351</td>
<td>0.35788</td>
<td>0.06485</td>
<td>0.10222</td>
</tr>
<tr>
<td>Mean error</td>
<td>0.09254</td>
<td>0.17199 ± 0.101</td>
<td>0.02835 ± 0.016</td>
<td>0.02681 ± 0.025</td>
</tr>
<tr>
<td><strong>Extrapolation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sample = 20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max error</td>
<td>0.10761</td>
<td>0.53594</td>
<td>0.09588</td>
<td>0.08595</td>
</tr>
<tr>
<td>Mean error</td>
<td>0.05797</td>
<td>0.23226 ± 0.165</td>
<td>0.05728 ± 0.031</td>
<td>0.04911 ± 0.027</td>
</tr>
</tbody>
</table>

*Note:* The results are obtained for 20 randomly chosen different initial weight values for ANNs and their mean values and standard deviations are given to show the minor effect of initial weight values on the results. Mean iteration corresponds to iterations needed to form the mapping \( P(.\) during the training phase.
of the process to obtain the model are increased. Hybrid models with less complexity can be proposed. For further work, another improvement would be reducing the convergence of SMN. In this study, as can be followed from applications the number of iterations during SMN parameter extraction is fixed for the SMN with PKI-D and CSMN methods. In other words, while for characteristic impedance modeling the iteration number is nine, it is three for the Branin function and these numbers of iterations are just the same number when SMN technique is used.

REFERENCES


AUTHORS’ BIOGRAPHIES

Murat Simsek

Murat Simsek was born on February 1, 1977 in Turkey. He received BSc degree from Yildiz Technical University (Istanbul), Faculty of Electrical and Electronics Engineering, Department of Electronics and Communication Engineering, Istanbul, in 1998. He received MSc degree from Istanbul Technical University, Institute of Science and Technology, Electronics and Communication Engineering Department in 2003. He accomplished his military obligation from 2003 to 2004. He attended PhD program in Istanbul Technical University, Institute of Science and Technology, Department of Electronics Engineering in 2004. He joined Istanbul Technical University, Faculty of Electrical and Electronics Engineering, Electronics and Communication Department, Circuits and Systems Division in 1999, where he is currently a Research Assistant. His research interests include space mapping based neuro-modeling, artificial neural networks, knowledge based neural networks and state minimization in incompletely specified sequential machine.
N. Serap Sengor received BSc, degree from Istanbul Technical University, Faculty of Electrical and Electronics Engineering, Electronics and Communication Engineering Department in 1985. She received MSc, and PhD degrees from Istanbul Technical University, Institute of Science and Technology, Electronics and Communication Engineering Department, in 1988 and 1995, respectively. She joined Istanbul Technical University, Faculty of Electrical and Electronics Engineering, Electronics and Communication Department as technical expert in 1986, where she is now Assistant Professor. She worked as Visiting Scientist at Circuit Theory Laboratory of Helsinki University of Technology in 2000–2001, and at Laboratory of Computational Engineering of Helsinki University of Technology in the 2006 summer. Her research interests include nonlinear circuits and systems, artificial neural networks and cognitive science.