A Microwave Filter Yield Optimization Method Based on Offline Surrogate Model Assisted Evolutionary Algorithm

Zhen Zhang, Bo Liu, *Senior Member, IEEE*, Yang Yu, *Student Member, IEEE*, Qingsha S. Cheng, *Senior Member, IEEE*

***Abstract*—Most existing microwave filter yield optimization methods target a small number of sensitive design variables (e.g., around 5). However, for many real-world cases, more than 10 sensitive design variables need to be considered. Due to the complexity, yield optimization quality and efficiency become challenges. Hence, a new method, called yield optimization for filters based on surrogate model-assisted evolutionary algorithm (YSMA), is proposed. The fundamental idea of YSMA is to construct a single high-accuracy surrogate model offline which fully replaces electromagnetic (EM) simulations in the entire yield optimization process. Global optimization is then enabled to find designs with substantial yield improvement efficiently using the surrogate model. To reduce the number of necessary samples (i.e., EM simulations) while obtaining the required prediction accuracy, a customized machine learning technique is proposed. The performance of YSMA is demonstrated by two real-world examples with 11 and 14 design variables, respectively. Experimental results show the advantages of YSMA compared to the current dominant sequential online surrogate model-based local optimization methods.**

***Index Terms*—Computationally expensive optimization, microwave filters, surrogate modeling, yield optimization.**

I. Introduction

MICROWAVE filters are key components in the front-end of wireless communication systems [1]. Different from many electromagnetic (EM) devices, filter responses are very sensitive to fabrication error. In other words, the performance of the fabricated filters often degrades significantly due to the fabrication error. As such, some of the fabricated filters are not qualified to be used. Yield is defined as the ratio of the number of qualified fabricated filters to the total number of fabricated filters considering fabrication error [2]. The higher the yield, the more robust the design, saving more fabrication and tuning costs.

Filter yield optimization starts from an existing design that already has good performance but may not be robust. The yield optimization algorithm then searches around this existing design, trying to fine tune it by maximizing yield [3]. The search range for yield optimization cannot be too small because a robust design near the existing design may be out of the very small search range, and thus, leads to little yield improvement. On the other hand, a large search range is also not necessary, because filter responses often degrade significantly even when a small deviation is added to the existing design parameters. Also, filter yield optimization does not aim at finding a robust design far away from the existing design. Hence, the search range is often decided empirically considering the above factors. In this paper, the search range is set to $\pm 0.05\%\lambda$, where λ is the wavelength of the center frequency of the targeted filter. (For example, considering a filter working at 10 GHz, the search range is ± 0.015 mm.) This empirical setting shows effectiveness for several tested real-world cases working at different frequencies. When adding a deviation that is more than $\pm 0.05\%\lambda$ to the existing design, samples satisfying the specification to be a qualified filter product can hardly be found.

A critical challenge for filter yield optimization is the speed of yield estimation. In most cases, to obtain an accurate filter response, EM simulation is needed, which is computationally expensive. The traditional way to estimate yield is the Monte-Carlo (MC) method, involving at least a few hundred EM simulations for a single candidate design. Despite that parallel computing is available, directly employing the traditional MC method is still unaffordable. Therefore, surrogate modeling methods are introduced and widely used [4], [5]. Surrogate models, which are often constructed by machine learning techniques, map the filter design parameters to the filter responses or features extracted from filter responses [6], [7]. Because they are computationally much cheaper than EM simulations, using surrogate models to estimate yield in the filter yield optimization process becomes a routine [7].

Widely used surrogate modeling methods in filter yield optimization include artificial neural networks (ANN) [7], [8], polynomial chaos (PC) [9]–[12] and Gaussian process (GP) [13]. Besides directly employing the above techniques, filter design knowledge is employed to improve the surrogate model quality. Important progress is the feature and transfer function-based method, e.g., [8], [14]–[16]. Features or transfer function parameters (e.g., poles, zeros, gains) are extracted from the response for surrogate modeling, largely reducing the complexity of the landscape to be modeled and shows clear

Manuscript received July 30th, 2021. This work was partially supported by the Scotland 5G Testbed Project 308106 and the National Natural Science Foundation of China Grant 62071211.

Corresponding authors: Bo Liu, Qingsha S. Cheng

Z. Zhang, Y. Yu and Q. S. Cheng are with Dept. of Electrical and Electronic Engineering, Southern University of Science and Technology, Shenzhen, 518055, P.R.China. (emails: 11849553@mail.sustech.edu.cn, issacyu@live.cn, chengqs@sustech.edu.cn).

B. Liu is with James Watt School of Engineering, University of Glasgow, Glasgow, Scotland. G12 8QQ. (e-mail: bo.liu@glasgow.ac.uk)

Yang Yu is also with the School of Electrical, Electronic and System Engineering, University of Birmingham, Birmingham B15 2TT, U.K. (e-mail: yxy726@student.bham.ac.uk).

prediction quality improvement.

However, most existing filter yield optimization methods target a small number of sensitive design variables (e.g., around 5) [8], [10]–[12]. In many real-world filters, the number of sensitive design variables is more than 10. Enabling efficient yield optimization for such cases is the aim of this paper. When handling more than 10 design variables, there are two main challenges for existing filter yield optimization methods.

First, most existing methods use a sequential or online surrogate modeling method [11], [17]. Specifically, a surrogate model is built within the fabrication error range for yield estimation. Once the algorithm iterates to a new candidate design, the surrogate model is rebuilt until convergence. When the number of sensitive design variables grows from a few to more than 10, the number of necessary EM simulations to build a reasonable surrogate model grows drastically (e.g., more than 15 times for PC modeling [18], Section III (A)). Hence, consecutive online surrogate model building leads to unaffordable time. Second, most existing methods use local optimization methods [10], [11]. Searching locally around an existing design, which is the aim of filter yield optimization, does not necessarily mean that the filter yield optimization problem is convex. On the contrary, our experiments show that when considering more than 10 sensitive design variables, the filter design landscape is multimodal and the quasi-Newton method [19] is often trapped in local optima, leading to insufficient yield improvement. Hence, yield optimization quality becomes an even more important challenge.

To address the above challenges, a new method, called yield optimization for filters based on surrogate model-assisted evolutionary algorithm (YSMA), is proposed. A single high-accuracy surrogate model for the entire yield optimization process is built offline. The necessary number of EM simulations becomes affordable. Differential evolution [20], carrying out global optimization, is then employed, which obtains designs with much more significant yield improvement than using local optimization. Building a high-accuracy surrogate model efficiently is the key to YSMA, and a customized machine learning technique is therefore proposed. Particularly, a method with quadratic support vector machine (SVM)-based classification and radio basis function neural network (RBFNN)-based regression using the extracted features is designed. Experiments and comparisons verify the advantage of YSMA.

The remainder of the paper is organized as follows. Section II presents the background knowledge. Section III elaborates on the YSMA method, including the workflow, the new customized surrogate modeling method, the optimization method, and parameter setting rules. Section IV presents the performance and advantages of YSMA using two real-world microwave filter examples with 11 and 14 sensitive design variables, respectively. The concluding remarks are provided in Section V.

II. Background Knowledge

A. Yield Estimation

Yield value describes the robustness of a given design $\boldsymbol{x} = [x_1, ..., x_d]$. In yield optimization, it is assumed that the given design $\boldsymbol{x}$ meets the design specifications without considering fabrication error. In machining, the fabrication error variable $\boldsymbol{\xi} = [\xi_1, ..., \xi_d]$ is added to $\boldsymbol{x}$. The exact value of $\boldsymbol{\xi}$ is not controllable and can only be represented as a random number following a certain distribution. For the real design $\boldsymbol{x} + \boldsymbol{\xi}$, when the specifications can still be met, the fabricated filter is qualified and is defined as $y_s(\boldsymbol{x}, \boldsymbol{\xi}) = 1$; Otherwise, $y_s(\boldsymbol{x}, \boldsymbol{\xi}) = 0$.

The yield of $\boldsymbol{x}$ is the probability of $y_s(\boldsymbol{x}, \boldsymbol{\xi}) = 1$, which is defined by:

$$P_r\left(y_s(\boldsymbol{x}, \boldsymbol{\xi}) = 1\right) = E\left[y_s(\boldsymbol{x}, \boldsymbol{\xi})\right] = \int y_s(\boldsymbol{x}, \boldsymbol{\xi})\rho(\boldsymbol{\xi})d\boldsymbol{\xi} \quad (1)$$

where $\rho(\boldsymbol{\xi})$ is the probability density function of $\boldsymbol{\xi}$.

In most cases, it is difficult to directly solve (1) analytically and the MC method [21] can be used as an estimation, which is defined as:

$$P_r\left(y_s(\boldsymbol{x}, \boldsymbol{\xi}) = 1\right) = \frac{1}{N_s}\sum_{i=1}^{N_s} y_s(\boldsymbol{x}, \xi_i) \quad (2)$$

where N_s is the total number of samples.

B. Vector Fitting and Feature Extraction

Vector fitting (VF) is a rational approximation method that is widely used in modeling and optimization of microwave devices [22]–[24]. Particularly, it is attracting attention in filter S-parameter response modeling and optimization in recent years. As shown in [15], [16], compared to directly modeling the S-parameters with respect to design variables by off-the-shelf machine learning methods, using VF to extract transfer function parameters to represent S-parameters obtains higher model quality. In this method, a reduced-order passive rational model to fit the frequency responses (e.g., S-parameters over frequency) as shown in (3) is obtained by VF and the accuracy is often high.

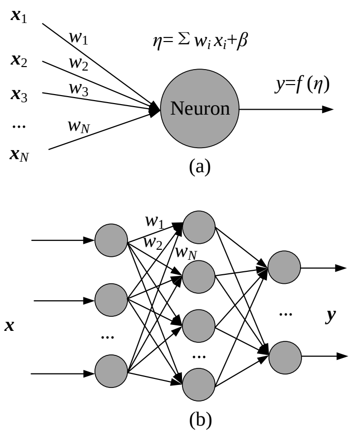
$$F(s) \approx \sum_{m=1}^{M} \frac{c_m}{s - a_m} + d + e \cdot s \quad (3)$$

where $s = 2\pi j \cdot f$ (f is the frequency).

VF is a pole relocation technique where the poles are improved by iteratively solving a linear problem until convergence. When it converges, the poles a_m, residuals c_m, terms d and e are obtained. In this paper, we call these parameters feature variables $\boldsymbol{h} = [a, c, d, e]$, which are used to represent the S-parameter response of a candidate filter design.

C. Artificial Neural Networks

ANN is widely used in microwave device design optimization [25], [26]. It has the ability of learning, association, memory, and fault tolerance, which is an effective method for classification and regression. Particularly, it shows good performance in modeling nonlinear functions [27]. The basic element of an ANN is a neuron, whose structure is shown in Fig. 1(a), where $w_i, i = 1, 2, ..., N$ are the weights, β is the bias, and $f(\eta)$ is the activation function.



(a)

(b)

Figure 1. Concepts of ANN. (a) A neuron and (b) A three-layer network.

The neurons can be linked together to construct a network as shown in Fig. 1(b). In the forward direction, each neuron receives data from neurons in the previous layer with different weights. A weighted sum is obtained, which is then transformed by its activation function to obtain the output of the neuron. The output serves as the input of neurons in the next layer. In this way, signals will propagate through the network layer-by-layer until the output layer. Backwardly, to minimize the error between the predicted outputs and the desired outputs, $\boldsymbol{w}$ and $\boldsymbol{\beta}$ can be optimized, which is called training. When the error is within an acceptable tolerance, the training terminates and the hyperparameters, i.e., $\boldsymbol{w}$ and $\boldsymbol{\beta}$, are decided. The ANN is then ready to be used for prediction. There are many kinds of ANNs whose performance can be very different for a targeted problem. In Section III, we will analyze the characteristics of the filter yield optimization problem and select a proper ANN type for it.

D. Support Vector Machines

Support vector machine (SVM) is another machine learning method, which is widely used in classification [28]. In YSMA, SVM is used as a binary classifier. SVM aims to find a hyperplane to segregate two categories. The criteria to define the best hyperplane are: (1) The objects belong to the correct category; (2) The margin (i.e., the distances between nearest data point (either category) and the hyperplane) is maximized. By solving the above constrained optimization problem, the hyperparameters to define the hyperplane can be obtained, which can be used to classify new objects.

When the categories to be segregated are not linearly separable, a kernel function can be introduced. The kernel function transforms a low-dimensional input space to a higher-dimensional space, and thus converts a non-separable problem to a separable problem. The hyperplane can then be obtained. More details are in [29].

E. Differential Evolution

Evolutionary algorithms have strength in handling multimodal optimization problems and are therefore widely used in global optimization [20]. Evolutionary algorithms mimic the process of natural selection. They often contain initialization, selection, genetic operators, and termination. These operators correspond to a certain facet of natural selection with the idea of fitter individuals will survive and produce, while unfit individuals will die off and have less chance to contribute to the gene pool of future generations. By iteratively carrying out this process, the fitness (i.e., optimality) of individuals in the population is improved and the optimal decision variables can eventually be obtained.

Differential evolution (DE) is arguably one of the currently most powerful evolutionary algorithms [20] and is used in microwave design optimization [6], [30]. In each iteration, it uses mutation, crossover, and selection operators. The mutation operator generates the difference vector(s) from selected individual vectors and adds it/them to other selected individual vectors so as to explore the search space. To enhance the population diversity, a crossover operation comes into play after mutation. The selection operator then determines the survived vector for the next generation based on a greedy selection criterion. In this work, DE is selected as the global optimizer for yield optimization.

III. The YSMA Algorithm

A. Challenges and Main Ideas of YSMA

Recall the two major challenges when handling filter yield optimization tasks with more than 10 sensitive design variables (Section I), which are efficiency and yield optimization quality. For the former, when the number of variables is more than 10, each yield evaluation involves a large number of EM simulations even employing surrogate models. For example, when using PC models [31] with 3 orders (i.e., good accuracy), the number of EM simulations needed for a single surrogate model is 868 when there are 10 variables, compared to 56 when there are 5 variables. Considering this, current dominant sequential (or online) surrogate model-assisted local optimization methods, requiring repeated surrogate modeling in optimization, may become unaffordable. For the latter, although the search space is small, the function landscape is highly multimodal based on our experiments (also shown in Section IV). Hence, local optimization is highly likely to be trapped in a local optimum, resulting in insufficient yield improvement. Carrying out global optimization is therefore necessary to obtain substantial yield improvement.

To address both challenges, a new framework is proposed. In contrast with the dominant method of sequentially building a set of surrogate models, a single offline high-accuracy surrogate model is built covering sufficient but not excessive search ranges. In contrast with the dominant local search-based yield optimization, global search is enabled. We assume that less than 5000 EM simulations in total for surrogate modeling are often affordable considering the current computing speed and resources. Also, the EM simulations for offline sampling can be performed in parallel.

Given the above, the central problem becomes how to meet the high accuracy requirement using a few thousand samples. This is not trivial because the filter design landscape is very

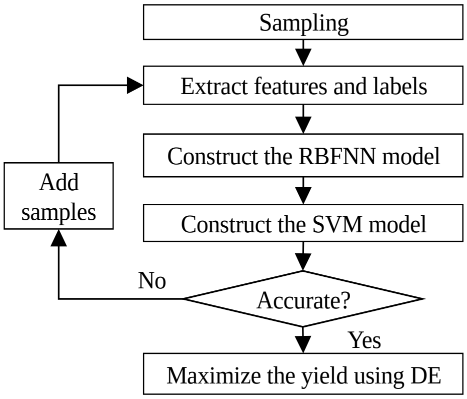

Sampling
Extract features and labels
Construct the RBFNN model
Construct the SVM model
Add samples
Accurate?
No
Yes
Maximize the yield using DE

Figure 2. Flow diagram of the YSMA algorithm.

sensitive to design parameters [6]. Thus, a customized machine learning method is proposed. Compared to most existing methods which regress between the design variables and S-parameters or transfer function parameters, we introduce a two-stage hybrid regression and classification method to construct the surrogate model. When the accuracy requirement is satisfied, DE will be used to carry out the global optimization. The flow diagram of YSMA is shown in Fig. 2.

Given an existing design $\boldsymbol{x}_0$, the modeling space $[\boldsymbol{a},\boldsymbol{b}]^d$ ($\boldsymbol{x}_0 \pm 0.05\%\lambda$ as recommended in Section I), and the fabrication error δ (e.g., 0.005 mm), the algorithm works as follows.

Step 1: Generate $1.1n \times d$ (d is the number of design variables) samples from the modeling space $[\boldsymbol{a},\boldsymbol{b}]^d$. Carry out EM simulations to obtain their S-parameters. Randomly select $n \times d$ samples as the training set, and $0.1n \times d$ samples as the test set.

Step 2: From the S-parameters of each sample in the training set, extract feature variables $\boldsymbol{h}$ using the VF method (Section II (A)), assign labels (i.e., 1/0) according to the $min(max(|S_{11}|))$ specification to define a qualified/not qualified product.

Step 3: Construct a RBFNN regression model between design variables ($\boldsymbol{x}$) and feature variables ($\boldsymbol{h}$) (Section III (B)).

Step 4: Construct a SVM classification model between feature variables ($\boldsymbol{h}$) and labels ($\boldsymbol{l}$) (Section III (B)). The hybrid model is $L(\boldsymbol{x}) = f_{SVM}(f_{RBF}(\boldsymbol{x}))$.

Step 5: Use the test set to evaluate the accuracy of the hybrid model. If the accuracy meets the threshold, go to Step 6. Otherwise, add $0.1n \times d$ training samples (including EM simulations) and go to Step 2.

Step 6: Use DE algorithm to obtain the robust design by maximizing the yield (Section III (C)). The search range is $[\boldsymbol{a}+\boldsymbol{\delta},\boldsymbol{b}-\boldsymbol{\delta}]^d$. In this process, the yield values are estimated by MC simulations and $L(\boldsymbol{x})$ is used to replace EM simulations.

B. Hybrid RBFNN-SVM Model for Yield Estimation

Because there is no EM simulation online, the surrogate model accuracy is the center of YSMA. Existing methods mostly use a feedforward multilayer perceptron neural network (MLPNN) to regress the design variables and the filter response [14], [32]. However, there might be two drawbacks: (1) Filter response sometimes degrades significantly (e.g., $max(|S_{11}|) = -5$ dB) when considering fabrication error, especially when there are more than 10 considered design variables. Hence, the output can be irregular: some performances are far from the specifications (i.e., very different response or feature variables compared to most other samples around the specifications), largely increasing the regression difficulty. (2) The MLPNN is a global approximation network and the output is controlled by all weights. Hence, several optimal settings (e.g., number of neurons in the hidden layer(s), activation functions) must be available together to obtain a high accuracy model, which is not easy. Being trapped in local optima in training and overfitting may also happen.

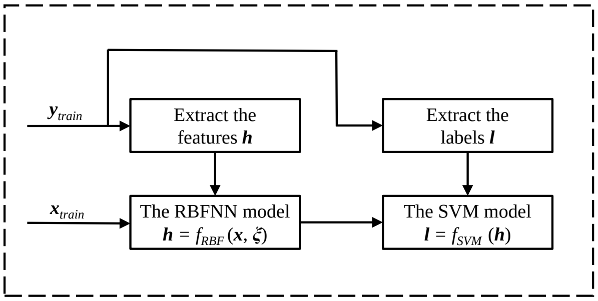

y_{train}
x_{train}
Extract the features $\boldsymbol{h}$
Extract the labels $\boldsymbol{l}$
The RBFNN model
$\boldsymbol{h} = f_{RBF}(\boldsymbol{x}, \xi)$
The SVM model
$\boldsymbol{l} = f_{SVM}(\boldsymbol{h})$

Figure 3. Flowchart of the hybrid surrogate model.

Hence, we propose a hybrid surrogate model as shown in Fig. 3. RBFNN maps the design variables $\boldsymbol{x}$ and feature variables $\boldsymbol{h}$. Note that due to the characteristics of the filter design landscape, identifying the correct type of ANN is essential considering the surrogate model quality. Using feature variables $\boldsymbol{h}$ alleviates the drawbacks of irregular data mentioned in the last paragraph, but the training data points are still not ideal for an MLPNN according to our experiments. Thus, RBFNN is selected considering the characteristics of the training data. Also, RBFNN is a local approximation network. By the RBF kernel, a local area of the input space is considered and thus, only a few weights affect the output. Moreover, the input space for filter yield optimization is very close to the existing design $\boldsymbol{x}_0$, which is small. Also, because only the mapping between the hidden layer and output layer needs to be trained, which is linear, the global optimum can be obtained. All of the above makes RBFNN have better accuracy than the widely used MLPNN for this particular problem.

To further avoid the effect of irregular output as mentioned above, regression is changed to classification. Hence, samples with largely degraded performance or those with performances just below the specifications are all labeled as 0. A powerful and robust classifier, SVM, is then selected to construct the overall model $L(\boldsymbol{x}) = f_{SVM}(f_{RBF}(\boldsymbol{x}))$. Implementation details are as follows.

A regularized RBFNN [33] is used, which has three layers. The weight between the input layer and hidden layer is 1. In the hidden layer, the number of neurons is the same as that of input samples. The activation function is RBF, which is shown in (4).

$$\varphi(\boldsymbol{x}) = exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{c}\|^2}{2\sigma^2}\right) \tag{4}$$

where $\boldsymbol{c}$ are radial basis centers. In regularized RBFNN, the radial basis centers are the input samples themselves. The width parameter σ is uniform to all neurons, which is $\sigma = d_m/\sqrt{2N}$, where d_m is the maximum distance between the radial basis centers, and N is the number of neurons in the hidden layer. The hidden layer maps the low-dimensional nonlinear separable inputs to the high-dimensional linear separable space for the output layer. It can be seen that when $\boldsymbol{x}_i$ is far away from $\boldsymbol{c}_j$, the output of the hidden layer neuron decays exponentially. The activation function of the neurons in the output layer is linear, and a weighted sum of the outputs of the hidden layer neurons form the final outputs (i.e., $\boldsymbol{h}$).

$$\begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1N} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ \varphi_{N1} & \varphi_{N2} & \cdots & \varphi_{NN} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \\ \vdots \\ \boldsymbol{w}_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \\ \vdots \\ \boldsymbol{h}_N \end{bmatrix} \tag{5}$$

where $\boldsymbol{w}_1$ is the weight vector with $1 \times N$ element.

$$\varphi_{ij} = \varphi\left(\|\boldsymbol{x}_j - \boldsymbol{c}_i\|\right), i, j = 1, \cdots, N \tag{6}$$

SVM is then used to construct a classification model from label pairs ($\boldsymbol{h}_i$,$\boldsymbol{l}_i$). A standard SVM model is used here, and the working principle is described in Section II (D). Based on our pilot experiments, the above filter classification problem is not linearly separable. Hence, a kernel function is introduced. Empirical analysis on various filters shows that quadratic polynomial kernel is among the competent ones, which is as follows.

$$K(\boldsymbol{h}_i, \boldsymbol{h}_j) = \phi(\boldsymbol{h}_i)^T\phi(\boldsymbol{h}_j) = (\boldsymbol{h}_i^T\boldsymbol{h}_j + r)^2 \tag{7}$$

where r is a polynomial constant.

Two parallel hyperplanes are then constructed, which are:

$$\boldsymbol{\varpi}^T\phi(\boldsymbol{h}) + b_s \geq +1 \tag{8}$$

$$\boldsymbol{\varpi}^T\phi(\boldsymbol{h}) + b_s \leq -1 \tag{9}$$

All the samples above the upper hyperplane belong to the qualified filters and vice versa. The distance between the two hyperplanes is defined as the margin $d_m = \frac{2}{\|\boldsymbol{\varpi}\|}$.

Then, d_m is maximized subject to fabricated filters are in correct categories to obtain the optimal hyperparameters. The hyperplanes can then be determined and the SVM model can be used for classification. The above constrained optimization problem is as follows.

$$\begin{aligned} \min_{\varpi, b_s} \quad & \frac{1}{2}\boldsymbol{\varpi}^T\boldsymbol{\varpi} + \kappa\sum\vartheta_i \\ \text{subject to} \quad & l_i\left(\boldsymbol{\varpi}^T\phi(\boldsymbol{h}_i)^T + b_s\right) \geq 1 - \vartheta_i \\ & \vartheta_i \geq 0 \end{aligned} \tag{10}$$

where $\boldsymbol{\varpi}$ is the normal vector to the hyperplane. κ>0 is the penalty coefficient of the error term. The slack variables ϑ_i refers to the degree of classification error of sample $\boldsymbol{h}_i$.

C. DE Algorithm for Yield Global Optimization

With the hybrid surrogate model to estimate yield accurately, EM simulations are fully replaced and efficiency no longer becomes a challenge. Global optimization can then be used. The maximization goal is the yield value, which is estimated by the hybrid surrogate model:

$$P_r\left(l(\boldsymbol{x}, \boldsymbol{\xi}) = 1\right) = \frac{1}{n}\sum_{i=1}^{n} f_{SVM}(f_{RBF}(\boldsymbol{x} + \boldsymbol{\xi}_i)) \tag{11}$$

DE is used as the global optimizer aiming to find the design variable $\boldsymbol{x}^*$ with the maximum estimated yield value. Note that other global optimization algorithms are also feasible. Our initial investigation shows that particle swarm optimization obtains slightly worse but comparable results with DE. Details of DE optimization are as follows.

In the Gth generation, the mutation is firstly carried out. Two candidate designs are subtracted to generate a difference vector, which is weighted and added to a third randomly selected candidate design. The result is a mutant vector $\boldsymbol{v}_{i,G}$.

$$\boldsymbol{v}_{i,G} = \boldsymbol{x}_{r_1,G} + F_s \cdot (\boldsymbol{x}_{r_2,G} - \boldsymbol{x}_{r_3,G}) \tag{12}$$

where the indices r_1 , r_2, and r_3 are mutually exclusive random integers within the population size. F_s is the scaling factor and refers to the differential vector weight.

After mutation, a trial vector $\boldsymbol{u}_{i,G}$ is formed by recombination of the mutation vector $\boldsymbol{v}_{i,G}$ and the target vector $\boldsymbol{x}_{i,G}$, which is also a candidate design. This is called the crossover operation. Binomial crossover is used here, which is as follows.

$$u_{j,i,G} = \begin{cases} v_{j,i,G} & \text{if } rand(0,1) \leq CR \\ x_{j,i,G} & \text{otherwise} \end{cases} \tag{13}$$

where CR is the crossover rate, j is the component index of a vector.

At last, selection takes place to choose the better one between the trial vector and the target vector according to the estimated yield values. The selection process is defined as follows:

$$\boldsymbol{x}_{i,G} = \begin{cases} \boldsymbol{u}_{i,G} & \text{if } y_s(\boldsymbol{u}_{i,G}) > y_s(\boldsymbol{x}_{i,G}) \\ \boldsymbol{x}_{i,G} & \text{otherwise} \end{cases} \tag{14}$$

D. Parameter Settings

YSMA has the following parameters: the modeling range, the number of samples to build the hybrid surrogate model, the order of VF, the threshold to accept the hybrid RBFNN and SVM model, and DE parameters. The modeling range is defined as $\pm 0.05\%\lambda$, where λ is the wavelength of the center frequency of the targeted filter. The reason for this setting is described in Section I. There are interesting research works studying the selection of subspace for surrogate modeling when considering yield optimization for antennas and microstrip circuits, e.g., [34], [35]. Our future work will include related methods for determining the search range.

In terms of the number of samples, we use $1.1n \times d$ as the initial value (Section III (A)). Using various filters working in different frequencies, the empirical setting is $n = 200$ and this number is not sensitive. Regarding the order of VF, starting from $2N_r+2$, where N_r is the number of resonators, the order of VF increases until both the average errors of S_{11} and S_{21} fitting (see (15)) are smaller than 0.3 dB.

Regarding the threshold, good accuracy is needed because no online EM simulation is carried out. On the other hand, a very high model accuracy requirement is not suggested. The model accuracy depends on the number of samples (i.e., simulation time) and the ability of the machine learning methods. Often, the initial design is obtained by 3D EM simulation-based design optimization without considering yield and the initial yield is often low. A surrogate model with reasonably high accuracy is sufficient when expecting sufficient yield improvement (e.g., increasing 30% to 80% instead of increasing 97% to 99.5%). Hence, the suggested overall model accuracy threshold is 85% considering the number of samples used (i.e., efficiency) and the ability of the machine learning method. For DE parameters, the setting is that the population size is $4 \times d$, $F_s = 0.8$, $CR = 0.5$, following [20].

IV. Results and Comparisons

In this section, two real-world filter examples are used to demonstrate the performance of YSMA. The first one is an X-band filter with 11 sensitive design variables and the second one is a C-band filter with 14 sensitive design variables after sensitivity analysis. To verify the advantages of our proposed hybrid surrogate modeling method, comparisons with popular machine learning methods are carried out. To verify the advantage of optimization quality, comparisons with the dominant quasi-Newton method are also carried out. Both examples are run on a PC with Intel 3.20 GHz Core (TM) i5 CPU and 20 GB RAM under Windows operating system. CST Microwave Studio is used as the EM simulator. Although no parallel computing is applied in the experiments, YSMA is straightforwardly compatible with parallel computing. All the time consumptions in the following are wall clock time.

A. Example 1

The first filter example is an X-band waveguide bandpass filter, which is shown in Fig. 4. The working frequency range is 9.8 GHz to 9.85 GHz. The section of the waveguide is 22.86 mm × 10.16 mm (WR-90). The sensitive design parameters are $\boldsymbol{x} = [l_1, l_2, l_3, l_4, l_5, k_{12}, k_{23}, k_{34}, k_{45}, q_{e1}, q_{e2}]$. The filter is modeled in CST Microwave Studio with about 27,000 meshes, and each simulation costs about 3 minutes. The existing design is $\boldsymbol{x}_0$=[18.950, 19.994, 19.817, 19.750, 18.770, 4.373, 4.377, 5.526, 4.987, 8.455, 8.948] (mm). The requirement for a fabricated filter to be qualified is $|S_{11}| \leq -16$ dB between 9.8 GHz to 9.85 GHz. The estimated yield value by $40 \times d$ EM simulations is 47.5%.

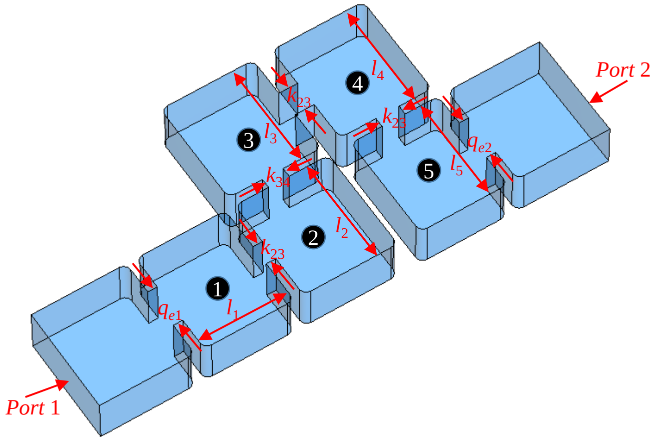


Figure 4. Example 1: the X-band bandpass filter.

YSMA is firstly employed. According to the parameter setting rules in Section III (D), 2420 samples are firstly used, where 2200 samples are used for training and 220 samples are used for testing, costing about 120 hours without parallel computing. Note that in filter development, testing and tuning are often the most time-consuming process, which could cost months [36], [37]. Compared to that, the efficiency of YSMA is sufficient even without parallel computing. When parallel computing is used (e.g., 10 computers in parallel), the efficiency can be considered reasonably high. In all the S-parameter-based accuracy tests in the following, the error is calculated by the following equation and the unit is dB.

$$\bar{E} = \frac{\sum_{k=1}^{K} \sum_{j=1}^{J} |\boldsymbol{S_p} - \boldsymbol{S_r}|}{K \cdot J} \tag{15}$$

where S_p is the predicted S-parameter, S_r is the simulated S-parameter at a certain frequency, J is the number of sampled frequency points, and K is the number of testing samples.

The $\bar{E}$ of VF (14 order) is less than –50 dB, which is highly accurate, and the number of dimensions of $\boldsymbol{h}$ is 59. RBFNN model is then constructed. The $\bar{E}$ for both S_{11} and S_{21} parameters are smaller than 0.2 dB for the RBFNN model. To show the advantages of the RBFNN for regressing $\boldsymbol{x}$ and $\boldsymbol{h}$, MLPNN is compared. As said in Section III (B), optimal settings for various hyperparameters are needed to obtain an accurate MLPNN. Different numbers of layers are tried and the optimal number of neurons are tuned manually. The following three MLPNNs show reasonably good performances: MLP_1 has 1 hidden layer with 15 neurons; MLP_2 has 2 hidden layers with 12 neurons in each layer; MLP_3 has 3 hidden layers and 10 neurons in each layer. Further increasing the number of layers does not improve the performance based on our trials. The comparison result is shown in Table I. Clear advantages of RBFNN can be observed. The predicted S-parameters of RBFNN and the best MLPNN (i.e., MLP_3) using a randomly selected test sample are shown in Fig. 5.

SVM is then used to classify labels ($\boldsymbol{l}$) from predicted features ($\boldsymbol{h}$). Here, classification accuracy refers to the probability of correct judgment, i.e., a qualified product is judged as

Table I
$\bar{E}$ Values of S-Parameters for RBFNN and MLPNN for Example 1 (in dB)

$\bar{E}$	$RBFNN$	MLP_1	MLP_2	MLP_3
S_{11}	0.113	6.071	5.887	3.954
S_{12}	0.147	22.173	20.498	15.232

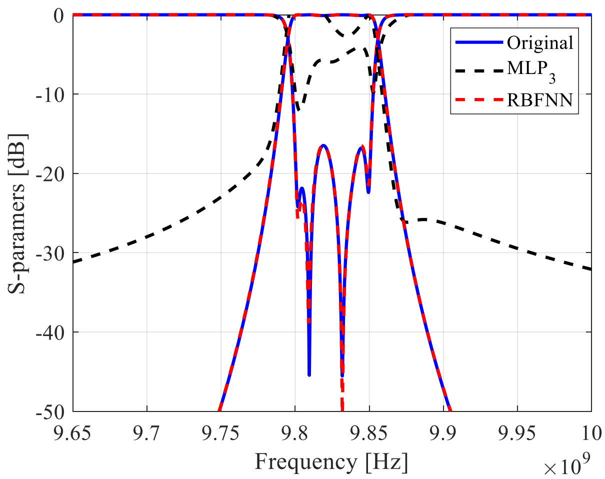


Figure 5. Predicted S-parameters using RBFNN and MLPNN for example 1.

qualified and a not qualified product is judged as not qualified. The classification accuracy is 92.6% using the test set. To verify the appropriateness of SVM for filter yield classification, popular machine learning methods are compared, including bagged trees [38], subspace discriminant [39] and subspace K-nearest neighbor methods [40]. The implementation is based on MATLAB Deep Learning Toolbox. Using the same training and test set, the comparison result is shown in Table II. It can be seen that SVM has good accuracy and outperforms the reference classification methods.

The overall accuracy (i.e., from design variables to judgment of qualified/unqualified products) comparisons with other widely used machine learning methods used in filter and other EM device (e.g., antenna, coupler) yield optimization [11], [34], [41] are carried out using the test set. Here, the accuracy still refers to the probability of correct judgment as before. The reference methods include PC and GP. The result is shown in Table III. It can be seen that the proposed hybrid model can obtain 95% accuracy and has significant advantages over PC and GP. Note that the accuracy between the SVM model and the hybrid model has a reasonable difference. This is due to the combined prediction error of the RBFNN and SVM models. The predicted yield of the existing design $\boldsymbol{x}_0$ is 47.5%, which is the same as that estimated by $40 \times d$ EM simulations.

Table II
Accuracy of Different Classification Methods for Example 1

	SVM	Bagged trees	Subspace discriminant	Subspace k-nearest neighbors
Accuracy (%)	92.6	89.5	84.4	76.6

Table III
Overall Accuracy of Different Machine Learning Methods for Example 1

	Hybrid RBFNN-SVM model	PC	GP
Accuracy (%)	95.0	88.7	84.3

Table IV
Yield Improvement Comparisons for Example 1

Methods	Initial yield	Optimized yield	Yield improvement
YSMA	47.5%	79%	31.5%
Quasi-Newton	47.5%	48%	0.5%
DE+PC model	47.5%	61.5%	14%
DE+GP model	47.5%	57%	9.5%

Since 85% is the threshold to accept the hybrid model, no further samples are needed, and DE optimization can be launched. For the yield estimation in the optimization process, $40 \times d$ samples are used, which are predicted by the hybrid surrogate model, so as to obtain the objective function value. Because no EM simulation is carried out online, DE optimization using 1000 generations costs less than 30 minutes. The optimal design with maximum predicted yield is $\boldsymbol{x}^*$=[18.941, 19.991, 19.817, 19.749, 18.770, 4.383, 4.387, 5.516, 4.977, 8.465, 8.958] (mm). The estimated yield is 82.5% compared to 79% using $40 \times d$ EM simulations, showing good accuracy.

Using the same hybrid model, a quasi-Newton optimizer is employed. The implementation is based on MATLAB Optimization Toolbox. The optimization converges to a local optimum with a yield of 48%. $40 \times d$ EM simulations are also used for each yield estimation in collaboration with the quasi-Newton method (i.e., no prediction error), which is very computationally expensive, and the final result is also 48%. This shows the contrast between global optimization and local optimization for filter yield optimization when considering more than 10 sensitive design variables. Methods using the same DE optimizer but combined with PC and GP surrogate models are also carried out. The comparison results are shown in Table IV. The effectiveness of the proposed hybrid model is shown because the power of the global optimizer is largely decreased when using PC and GP surrogate models.

B. Example 2

The second filter example is an C-band waveguide bandpass filter, which is shown in Fig. 6. The working frequency range is 4.9 GHz to 5.1 GHz. The section of the waveguide is 42.00 mm × 17.00 mm (WR-42). The sensitive design variables are $\boldsymbol{x}$=[Q_{e1}, Q_{e2}, L_1, L_2, L_3, L_4, L_5, L_6, K_{12}, K_{23}, K_{34}, K_{45}, K_{56}, K_{25}] which are shown in the figure. The filter is modeled in CST Microwave Studio with about 14,000 meshes, and each simulation costs about 2 minutes. The existing design is $\boldsymbol{x}_0$=[2.984, 2.802, 50.262, 86.599, 41.451, 43.830, 43.167, 50.004, 2.702, 3.308, 21.261, 2.925, 2.432, 12.036] (mm). The requirement for a fabricated filter to be qualified is $|S_{11}| \leq -16$ dB between 4.9 GHz to 5.1 GHz. The estimated yield value by $40 \times d$ EM simulations is 56%.

YSMA is firstly employed. According to the parameter setting rules in Section III (D), 3080 samples are firstly used,

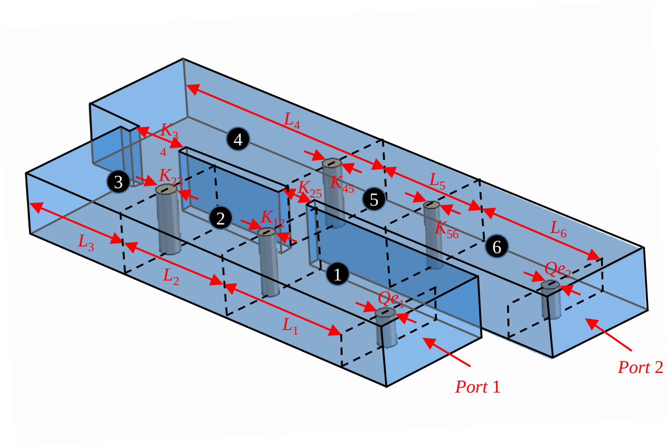


Figure 6. Example 2: the C-band bandpass filter.

Table V
$\bar{E}$ Values of S-Parameters for RBFNN and MLPNN for Example 2 (in dB)

$\bar{E}$	$RBFNN$	MLP_{best}
S_{11}	0.280	17.557
S_{12}	0.124	23.407

where 2800 samples are used for training and 280 samples are used for testing. However, the accuracy of the hybrid RBFNN-SVM model is 83%. Hence, two infilling of $20 \times d$ each (Step 5, Section III (A)) are performed until the accuracy is higher than 85%, which is the threshold. Therefore, 3640 samples are finally used, costing about 120 hours without parallel computing.

The $\bar{E}$ of VF (18 order) is less than -60 dB, which is highly accurate, and the number of dimensions of $\boldsymbol{h}$ is 75. RBFNN model is then constructed. The $\bar{E}$ for both S_{11} and S_{21} parameters are smaller than 0.3 dB for the RBFNN model. MLPNN is also compared. As in example 1, different numbers of layers are tried and the optimal number of neurons are tuned manually. The best one has 1 hidden layer and 15 neurons. The comparison result is shown in Table V. Clear advantages of RBFNN can be observed. The predicted S-parameters of RBFNN and the best MLPNN using a randomly selected test sample are shown in Fig. 7.

Table VI
Accuracy of Different Classification Methods for Example 2

	SVM	Bagged trees	Subspace discriminant	Subspace k nearest neighbor
Accuracy (%)	91.4	87.1	84.7	52.6

Table VII
Overall Accuracy of Different Machine Learning Methods for Example 2

	Hybrid RBFNN-SVM model	PC	GPR
Accuracy (%)	85.6	78.5	75.0

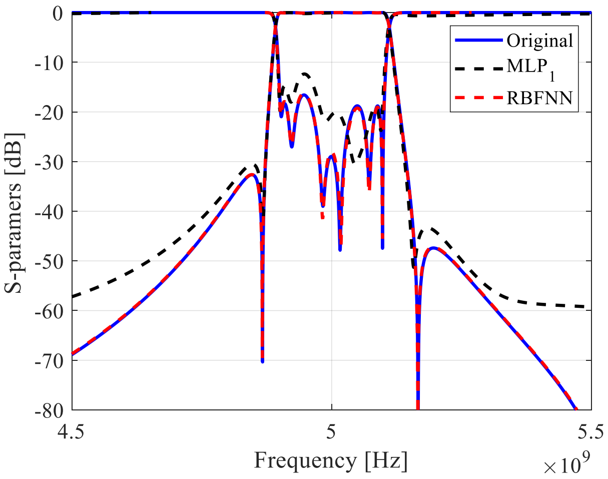


Figure 7. Predicted S-parameters using RBFNN and MLPNN for example 2.

Table VIII
Yield Improvement Comparisons for Example 2

Methods	Initial yield	Optimized yield	Yield improvement
YSMA	56%	97.2%	41.2%
Quasi-Newton	56%	62.2%	6.2%
DE+PC model	56%	85%	29%
DE+GP model	56%	84%	28%

In terms of verifying the advantages of the SVM classifier, the same comparison as example 1 is carried out and the comparison result is shown in Table VI. The classification accuracy of SVM is 91.4% and an advantage is shown compared to other reference methods.

For the overall accuracy, like example 1, the hybrid model is compared with PC and GP using the same training and test sets. The result is shown in Table VII. It can be seen that the proposed hybrid model can obtain 85.7% accuracy and has significant advantages over PC and GP. The difference between SVM classification accuracy and overall accuracy is also due to the combined prediction error of the RBFNN and SVM. The predicted yield of the existing design x_0 is 56%, compared to 56.6% by $40 \times d$ EM simulations.

DE optimization is at last launched. All the settings are the same as example 1, costing less than 30 minutes. The optimal design with maximum predicted yield is $\boldsymbol{x}^{*}$=[2.976, 2.800, 50.282, 43.781, 41.431, 86.684, 43.170, 49.997, 2.701, 3.294, 21.281, 2.927, 2.422, 12.028] (mm). The estimated yield is 100% compared to 97.2% using $40 \times d$ EM simulations, showing good accuracy. Using the same hybrid model, quasi-Newton optimization is compared. The optimization converges to a local optimum with a yield of 62.2%. The comparison results are shown in Table VIII. This again shows the necessity of introducing global optimization when considering more than 10 sensitive design variables in filter yield optimization. Because using $40 \times d$ EM simulations in collaboration with the quasi-Newton method is computationally too expensive, the experiment in example 1 is not repeated. In terms of the effectiveness of the proposed hybrid model in contrast with PC and GP surrogate models, the same observation can be made as example 1 according to Table VIII.

V. Conclusions

In this paper, the YSMA algorithm has been proposed. To the best of our knowledge, YSMA is the first method for filter yield optimization considering more than 10 sensitive design variables. Experiments show significant yield improvement (i.e., 30% to 40%) within a practical timeframe as well as advantages compared to the current dominant filter yield optimization methods. The effectiveness of YSMA comes from the proposed customized machine learning method for this particular problem, which is also compatible with other filter optimization techniques, and the introduction of global optimization into this research topic. Future works include investigating a more systematic method for defining the surrogate modeling and search range by adapting the existing interesting related research.

Acknowledgement

The authors would like to thank Prof. Muhammad Imran, University of Glasgow, Scotland, for valuable discussions and contributions.

References

[1] J.-S. G. Hong and M. J. Lancaster, *Microstrip filters for RF/microwave applications*. John Wiley & Sons, USA, 2004, vol. 167.

[2] A. Chipouras, S. Markatos, T. Sphicopoulos, C. Caroubalos, H. Mahlein, and R. Matz, "Performance estimation and yield analysis of an oeic passive duplexer," *Journal of Lightwave Technology*, vol. 14, no. 2, pp. 164–168, 1996.

[3] J. W. Bandler and S. H. Chen, "Circuit optimization: The state of the art," *IEEE Transactions on Microwave Theory and Techniques*, vol. 36, no. 2, pp. 424–443, 1988.

[4] S. Koziel, D. E. Ciaurri, and L. Leifsson, *Surrogate-based methods*. Springer, Germany, 2011.

[5] S. Koziel, A. Bekasiewicz, P. Kurgan, and J. W. Bandler, "Expedited multi-objective design optimization of miniaturized microwave structures using physics-based surrogates," in *2015 IEEE MTT-S International Microwave Symposium, Phoenix, AZ, USA*, 2015, pp. 1–3.

[6] B. Liu, H. Yang, and M. J. Lancaster, "Global optimization of microwave filters based on a surrogate model-assisted evolutionary algorithm," *IEEE Transactions on Microwave Theory and Techniques*, vol. 65, no. 6, pp. 1976–1985, 2017.

[7] J. E. Rayas-Sanchez and V. Gutierrez-Ayala, "Em-based monte carlo analysis and yield prediction of microwave circuits using linear-input neural-output space mapping," *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 12, pp. 4528–4537, 2006.

[8] J. Zhang, F. Feng, J. Jin, W. Zhang, and Q. J. Zhang, "Adaptively weighted yield-driven em optimization incorporating neurotransfer function surrogate with applications to microwave filters," *IEEE Transactions on Microwave Theory and Techniques*, vol. 69, no. 1, pp. 518–528, 2021.

[9] G. D'Antona, A. Monti, F. Ponci, and L. Rocca, "Maximum entropy analytical solution for stochastic differential equations based on the wiener-askey polynomial chaos," in *Proceedings of the 2006 IEEE International Workshop on Advanced Methods for Uncertainty Estimation in Measurement (AMUEM 2006), Sardinia, Italy*, 2006, pp. 62–66.

[10] J. Zhang, F. Feng, W. Na, S. Yan, and Q. Zhang, "Parallel space-mapping based yield-driven em optimization incorporating trust region algorithm and polynomial chaos expansion," *IEEE Access*, vol. 7, pp. 143 673–143 683, 2019.

[11] J. Zhang, C. Zhang, F. Feng, W. Zhang, J. Ma, and Q. J. Zhang, "Polynomial chaos-based approach to yield-driven em optimization," *IEEE Transactions on Microwave Theory and Techniques*, pp. 3186–3199, 2018.

[12] Z. Zhang, H. Chen, Y. Yu, F. Jiang, and Q. S. Cheng, "Yield-constrained optimization design using polynomial chaos for microwave filters," *IEEE Access*, vol. 9, pp. 22 408–22 416.

[13] A. Pietrenko-Dabrowska, S. Koziel, and M. Al-Hasan, "Expedited yield optimization of narrow- and multi-band antennas using performance-driven surrogates," *IEEE Access*, vol. 8, pp. 143 104–143 113, 2020.

[14] F. Feng, W. Na, W. Liu, S. Yan, L. Zhu, J. Ma, and Q. Zhang, "Multifeature-assisted neuro-transfer function surrogate-based em optimization exploiting trust-region algorithms for microwave filter design," *IEEE Transactions on Microwave Theory and Techniques*, vol. 68, no. 2, pp. 531–542, 2020.

[15] Y. Cao, G. Wang, and Q.-J. Zhang, "A new training approach for parametric modeling of microwave passive components using combined neural networks and transfer functions," *IEEE Transactions on Microwave Theory and Techniques*, vol. 57, no. 11, pp. 2727–2742, 2009.

[16] F. Feng, C. Zhang, J. Ma, and Q.-J. Zhang, "Parametric modeling of em behavior of microwave components using combined neural networks and pole-residue-based transfer functions," *IEEE Transactions on Microwave Theory and Techniques*, vol. 64, no. 1, pp. 60–77, 2015.

[17] H. L. Abdel-Malek, A. S. O. Hassan, E. A. Soliman, and S. A. Dakroury, "The ellipsoidal technique for design centering of microwave circuits exploiting space-mapping interpolating surrogates," *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, pp. 3731–3738, 2006.

[18] Z. Zhang, T. A. El-Moselhy, I. M. Elfadel, and L. Daniel, "Stochastic testing method for transistor-level uncertainty quantification based on generalized polynomial chaos," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 10, pp. 1533–1545, 2013.

[19] Z. Li, W. Wu, B. Zhang, and H. Sun, "Dynamic economic dispatch using lagrangian relaxation with multiplier updates based on a quasi-newton method," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4516–4527, 2013.

[20] K. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.

[21] Y. Li, H. Schneider, F. Schnabel, R. Thewes, and D. Schmitt-Landsiedel, "Dram yield analysis and optimization by a statistical design approach," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 12, pp. 2906–2918, 2011.

[22] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by vector fitting," *IEEE Transactions on Power Delivery*, vol. 14, no. 3, pp. 1052–1061, 1999.

[23] S. Grivet-Talocia and M. Bandinu, "Improving the convergence of vector fitting for equivalent circuit extraction from noisy frequency responses," *IEEE Transactions on Electromagnetic Compatibility*, vol. 48, no. 1, pp. 104–120, 2006.

[24] D. Deschrijver, M. Mrozowski, T. Dhaene, and D. D. Zutter, "Macromodeling of multiport systems using a fast implementation of the vector fitting method," *IEEE Microwave and Wireless Components Letters*, vol. 18, no. 6, pp. 383–385, 2008.

[25] A. H. Zaabab and Q. J. Zhang, "A neural network modeling approach to circuit optimization and statistical design," *IEEE Transactions on Microwave Theory and Techniques*, vol. 43, no. 6, pp. 1349–1358, 1995.

[26] J. Jin, C. Zhang, F. Feng, W. Na, J. Ma, and Q. J. Zhang, "Deep neural network technique for high-dimensional microwave modeling and applications to parameter extraction of microwave filters," *IEEE Transactions on Microwave Theory and Techniques*, vol. 67, no. 10, pp. 4140–4155, 2019.

[27] J. Dalton and A. Deshmane, "Artificial neural networks," *IEEE Potentials*, vol. 10, no. 2, pp. 33–36, 1991.

[28] S. Theodoridis and M. Mavroforakis, "Reduced convex hulls: A geometric approach to support vector machines," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 119–122, 2007.

[29] S. Pang, D. Kim, and S. Y. Bang, "Face membership authentication using svm classification tree generated by membership-based lle data partition," *IEEE Transactions on Neural Networks*, vol. 16, no. 2, pp. 436–46, 2005.

[30] B. Liu, H. Yang, and M. J. Lancaster, "Synthesis of coupling matrix for diplexers based on a self-adaptive differential evolution algorithm," *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 2, pp. 813–821, 2018.

[31] S. L. Ho and S. Yang, "A fast robust optimization methodology based on polynomial chaos and evolutionary algorithm for inverse problems," *IEEE Transactions on Magnetics*, vol. 48, no. 2, pp. 259–262, 2012.

[32] C. Zhang, F. Feng, V. M. R. Gongal-Reddy, Q. J. Zhang, and J. W. Bandler, "Cognition-driven formulation of space mapping for equal-ripple optimization of microwave filters," *IEEE Transactions on Microwave Theory and Techniques*, vol. 63, no. 7, pp. 2154–2165, 2015.

[33] P. V. Yee and S. Haykin, *Regularized Radial Basis Function Networks: Theory and Applications*. John Wiley, USA, 2001.

[34] A. Pietrenko-Dabrowska, S. Koziel, and M. Al-Hasan, "Expedited yield optimization of narrow- and multi-band antennas using performance-driven surrogates," *IEEE Access*, vol. 8, pp. 143 104–143 113, 2020.

[35] S. Koziel, A. Pietrenko-Dabrowska, Q. S. Cheng, and Z. Zhang, "Low-cost surrogate modeling of compact microstrip circuits in highly-dimensional parameters spaces using variable-fidelity nested co-kriging," in *2020 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO), Hangzhou, China*. IEEE, 2020, pp. 1–4.

[36] N. Zahirovic, R. R. Mansour, and M. Yu, "Scalar measurement-based algorithm for automated filter tuning of integrated chebyshev tunable filters," *IEEE Transactions on Microwave Theory and Techniques*, vol. 58, no. 12, pp. 3749–3759, 2010.

[37] M. M. Inc., "Quality and reliability report," 215 Vineyard Court, Morgan Hill, CA., Tech. Rep., 2018.

[38] P. K. Mishra, A. Yadav, and M. Pazoki, "A novel fault classification scheme for series capacitor compensated transmission line based on bagged tree ensemble classifier," *IEEE Access*, vol. 6, pp. 27 373–27 382, 2018.

[39] S. Zhang and T. Sim, "Discriminant subspace analysis: A fukunaga-koontz approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, pp. 1732–1745, 2007.

[40] B. S. Kim and S. B. Park, "A fast k nearest neighbor finding algorithm based on the ordered partition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 761–766, 1986.

[41] S. Koziel and A. Bekasiewicz, "Variable-fidelity response feature surrogates for accelerated statistical analysis and yield estimation of compact microwave components," *IET Microwaves Antennas & Propagation*, vol. 13, no. 14, 2019.

Zhen Zhang received the bachelor's degree from Anhui University of Science and Technology, China, in 2015, and the master's degree from Northwest A & F University, China, in 2018. She is currently pursuing the Ph.D. degree with the Southern University of Science and Technology (SUSTech), Shenzhen, China and the Harbin Institute of Technology, Harbin, China. Her research interests include antenna design optimization, rapid design optimization method of microwave components, uncertainty analysis, and yield optimization.

Bo Liu (M'15-SM'17) received the B.S. degree from Tsinghua University, P. R. China, in 2008. He received his Ph.D. degree at University of Leuven (KU Leuven), Belgium, in 2012. From 2012 to 2013, he was a Humboldt research fellow and was working with Technical University of Dortmund, Germany. In 2013, he was appointed Lecturer at Wrexham Glyndwr University, UK, where he was promoted to Reader in 2016. Since 2020, he is a Senior Lecturer at University of Glasgow, UK. He is also an honorary fellow at University of Birmingham, UK. His research interests lie in AI-driven design methodologies of analog/RF integrated circuits, microwave devices, MEMS, evolutionary computation and machine learning. He has authored or co-authored 1 book and more than 70 papers in renowned international journals, edited books and conference proceedings. He is the technical leader of this microwave filter yield optimization project.

Yang Yu was born in Tianjin, P.R.China, in 1991. He received the B.Eng. degree in communication engineering and the M.Eng. degree in information and communication engineering at Tianjin Polytechnic University, Tianjin, P.R.China, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree at University of Birmingham, Birmingham, U.K.. He is now a joint Ph.D. student with the University of Birmingham, Birmingham, U.K., and Southern University of Science and Technology, Shenzhen, P.R.China. His recent research focuses on synthesis and design of RF/microwave components and computational intelligence techniques for engineering.

Qingsha S. Cheng (S'00–M'05–SM'09) received the B.Eng. and M. Eng. from Chongqing University, China, in 1995 and 1998, respectively. He received his Ph.D. at McMaster University, Canada, in 2004. In 1998, he was with the Department of Computer Science and Technology, Peking University, China. In 2004, he became a postdoctoral fellow and then a research engineer in 2007, both with the Department of Electrical and Computer Engineering, McMaster University. In 2014, he joined Southern University of Science and Technology (SUSTech), Shenzhen, China as a faculty member in the Department of Electrical and Electronic Engineering. His research interests include smart modelling and optimization of microwave components and antennas, surrogate modelling and optimization, multi-objective optimization, and machine learning. He has authored or co-authored more than 190 publications in technical book chapters, refereed international technical journals, refereed international conference proceedings and international workshops. His works have been cited more than 3200 times according to Google scholar.