Machine Learning Based PCB/Package Stack-up Optimization for Signal Integrity

Wenchang Huang
Missouri University of Science and Technology
Rolla, USA
wh3vw@umsystem.edu

Jiahuan Huang
Missouri University of Science and Technology
Rolla, USA
jhkmm@umsystem.edu

Minseok Kim
Samsung Electronics Co., Ltd
Suwon, South Korea
mixerk.kim@samsung.com

Bumhee Bae
Samsung Electronics Co., Ltd
Suwon, South Korea
bh1.bae@samsung.com

Subin Kim
Samsung Electronics Co., Ltd
Suwon, South Korea
subinn.kim@samsung.com

Chulsoon Hwang
Missouri University of Science and Technology
Rolla, USA
hwangc@umsystem.edu

2023 IEEE 73rd Electronic Components and Technology Conference (ECTC) | 979-8-3503-3498-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ECTC51909.2023.00323

***Abstract*— PCB/package stack-up design optimization is time-consuming and requiring a great deal of experience. Although some iterative optimization algorithms are applied to implement automatic stack-up design, evaluating the results of each iteration is still time-intensive. This paper proposes a combined Bayesian optimization-artificial neural network (BO-ANN) algorithm, utilizing a trained ANN-based surrogate model to replace a 2D cross-section analysis tool for fast PCB/package stack-up design optimization. With the acceleration of ANN, the proposed BO-ANN algorithm can finish 100 iterations in 40 seconds while achieving the target characteristic impedance. To better generalize the BO-ANN algorithm, a strategy of effective dielectric calculation is applied to multiple-dielectric stack-up optimization. the BO-ANN algorithm will be able to output optimized stack-up designs with dielectric layers chosen from the pre-defined library and the obtained designs are verified by 2D solver.**

Keywords—PCB/package stack-up design, artificial neural network, deep learning, Bayesian optimization

I. Introduction

The electrical properties of the transmission line are usually considered for PCB/package stack-up design, such as impedance matching and attenuation minimization. Even though designers can obtain the transmission line electrical properties by putting stack-up parameters into commercially available 2D cross-section analysis tools, the mapping from stack-up parameters to transmission line electrical properties is non-linear. Therefore, design optimization still requires lots of time and experience. Especially when multiple dielectrics are applied in PCB/package stack-up design, the complexity of design optimization increases dramatically.

Several works have applied machine learning based optimization on the design process as a substitute for manual selection. One popular method is the genetic algorithm (GA), which is widely used for diverse problems in signal integrity/power integrity [1][2]. However, GA requires many evaluations of the optimization function, which limits its application. Another widely used optimization technique for SI/PI is deep reinforcement learning [3]. Whereas, it is hard to find an optimal solution with a large variable space. In recent years, non-convex Bayesian optimization (BO) has become popular. BO is a Gaussian process (GP) and a Bayes-theorem-based approach, which aims to optimize N-dimensional functions and minimize the number of evaluations [4]. For SI/PI tasks, BO is used to optimize via parameters for signal transmission [5], the eye opening of interconnects in high-speed channels [6], and PCB stack-up [7]. BO works well in [5], however, the evaluation is still expensive. To further accelerate the algorithm, the additive Gaussian process (ADD-GP) [8] is used as a predictive model in [6] while each output needs one ADD-GP model. Although [7] utilizes an artificial neural network to replace the 2D cross-section analysis tool, the simple structure of the neural network and small dataset cannot guarantee prediction accuracy. Furthermore, only a 2-dielectric stack-up with 5 parameters is studied in [7], this strategy cannot solve the problem of data explosion caused by multiple dielectrics and the difficulty of convergence in a large variable space.

In this paper, the BO-ANN algorithm is proposed with an effective dielectric calculation strategy for fast PCB/package stack-up optimization. The variables are the stack-up parameters of the PCB/package, and the objective function is defined by the electrical properties of the PCB/package: the characteristic impedance (Z_0) and attenuation. The ANN is introduced to calculate the objective function instead of time-consuming electromagnetic simulations. After training ANN on over 100,000 stack-ups, the relative prediction error of Z_0 is less than 1% on the validation dataset, and the relative prediction error of attenuation is less than 5%. Furthermore, the difference between the Z_0 of the optimized stack-up and the target Z_0 is less than 2% on practical applications, and the difference of attenuation is less than 11%.

II. Bayesian optimization approach

BO was originally developed in the 1970s and 1980s [9]-[11]. In this paper, BO is the main frame of the proposed BO-ANN algorithm, and the flow chart is shown in Fig. 1. Different from conventional BO, the ANN model acts as an evaluator to accelerate algorithm speed. Initially, BO uses GP

This work was supported in part by the National Science Foundation (NSF) under Grant IIP-1916535

2377-5726/23/$31.00 ©2023 IEEE
DOI 10.1109/ECTC51909.2023.00323

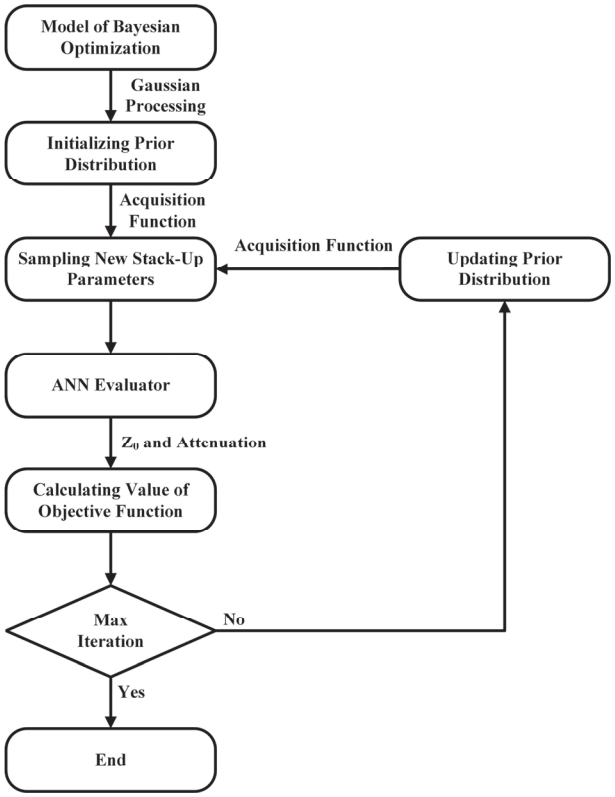


Fig. 1. Flow chart of the proposed algorithm.

to estimate parameter possibility distributions and relationships by several known data points. Based on the fitted parameter joint probability distribution, the acquisition function (AF) will determine the potential optimal point as the next sample point, then update the prior joint probability distribution with the new sample point. Accordingly, BO consists of two important parts, GP and AF.

A. Multivariate Gaussian Process

For PCB/package stack-up optimization, there are multiple parameters to be optimized. The multivariate Gaussian distribution is considered in this multidimensional problem. First, one-dimensional Gaussian distribution [12] is given by:

$$X \sim N(\mu,\sigma^2) \tag{1}$$

Where μ is the mean and σ is the standard deviation. The variance of the distribution is σ^2. And the probability density function for one-dimensional Gaussian distribution is given as follows:

$$p(x)=\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{2}$$

Similarly for multidimensional Gaussian distribution [13], it is notated as (3), and the probability density function in (4):

$$\boldsymbol{X} \sim N(\boldsymbol{\mu},\boldsymbol{\Sigma}) \tag{3}$$

$$f_X(X_1,\ldots,X_N)=\frac{1}{\sqrt{(2\pi)^N|\boldsymbol{\Sigma}|}}\exp\left(-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{X}-\boldsymbol{\mu})\right) \tag{4}$$

where $\boldsymbol{X}=(X_1,\ldots,X_N)$ is the n-dimension random vector, $\boldsymbol{\mu}=(\mu_1,\ldots,\mu_N)$ is the mean vector, and $\boldsymbol{\Sigma}$ is the covariance matrix of size $N\times N$.

The GP is the joint distribution of all those random variables. For example, the collection of those random variables has a multidimensional Gaussian distribution. For a random vector $\boldsymbol{u}\in\mathbf{R}^{\boldsymbol{n}}$ with $\boldsymbol{u}\sim N(\boldsymbol{\mu},\boldsymbol{\Sigma})$, the covariance matrix $\boldsymbol{\Sigma}$ describes the relationship between variables. The kernel function is used to define the covariance matrix that will be discussed in the following part. It is supposed that the variable u is separated to $\boldsymbol{u}_A=(u_1,\ldots,u_r)$ and $\boldsymbol{u}_B=(u_{r+1},\ldots,u_n)$, similarly, for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, such that:

$$\boldsymbol{u}=\begin{bmatrix}\boldsymbol{u}_A\\ \boldsymbol{u}_B\end{bmatrix}$$

$$\boldsymbol{\mu}=\begin{bmatrix}\boldsymbol{\mu}_A\\ \boldsymbol{\mu}_B\end{bmatrix}$$

$$\boldsymbol{\Sigma}=\begin{pmatrix}\boldsymbol{\Sigma}_{AA} & \boldsymbol{\Sigma}_{AB}\\ \boldsymbol{\Sigma}_{BA} & \boldsymbol{\Sigma}_{BB}\end{pmatrix} \tag{5}$$

And for the separate random vectors, $\boldsymbol{u}_A$ and $\boldsymbol{u}_B$ also follows Gaussian Distribution, such that:

$$\boldsymbol{u}_A \sim N(\boldsymbol{\mu}_A,\boldsymbol{\Sigma}_{AA}) \tag{6}$$

$$\boldsymbol{u}_B \sim N(\boldsymbol{\mu}_B,\boldsymbol{\Sigma}_{BB}) \tag{7}$$

The conditional probability distribution of $\boldsymbol{u}_B$ given $\boldsymbol{u}_A$ is also a Gaussian distribution given by:

$$\boldsymbol{u}_B|\boldsymbol{u}_A \sim N\left(\boldsymbol{\mu}_B+\boldsymbol{\Sigma}_{BA}\boldsymbol{\Sigma}_{AA}^{-1}(\boldsymbol{u}_A-\boldsymbol{\mu}_A),\boldsymbol{\Sigma}_{BB}-\boldsymbol{\Sigma}_{BA}\boldsymbol{\Sigma}_{AA}^{-1}\boldsymbol{\Sigma}_{AB}\right) \tag{8}$$

In BO, The Gaussian regression process is used to find the relations between variables and the objective function in training data. First, a group of sampling points is known, $a_{1:t}=(a_1,\ldots,a_t)$ and $f(a_{1:t})=(f(a_1),\ldots,f(a_t))$, and it follows Joint GP with $\mu(a_{1:t})$ as mean and $\boldsymbol{K}$ as a covariance matrix. $\boldsymbol{K}$ is calculated with a kernel function. As for a new point, it still follows the distribution, such that:

$$\begin{bmatrix}f(a_{1:t})\\ f(a_{t+1})\end{bmatrix}\sim N\left(\begin{bmatrix}\mu(a_{1:t})\\ \mu(a_{t+1})\end{bmatrix},\begin{bmatrix}\boldsymbol{K} & \boldsymbol{k}\\ \boldsymbol{k}^T & k(a_{t+1},a_{t+1})\end{bmatrix}\right) \tag{9}$$

It assumes that a prior distribution is given which is from the known sampling points, $a_{1:t}$, and the posterior predictive distribution of the next sampling point is given as follows:

$$f(a_{t+1})\big|f(a_{1:t}) \sim N(\mu,\sigma^2)$$

$$\mu = \boldsymbol{k}^T \boldsymbol{K}^{-1}\left(f(a_{1:t}) - \mu(a_{1:t})\right) + \mu(a_{t+1})$$

$$\sigma^2 = k(a_{t+1},a_{t+1}) - \boldsymbol{k}^T \boldsymbol{K}^{-1}\boldsymbol{k} \tag{10}$$

The often-used kernel function includes linear function, Gaussian noise function, squared exponential function, Matern function, and rational quadratic function [14]. The Matern 5/2 function is used in this work. Different kernel functions used in the BO for stack-up design are discussed in detail in [7]. In BO, Gaussian regression process gives the probability distribution of the objective function f based on the known sampling points $f(a_{1:t})$ and quantifies the uncertainty of unknown areas.

B. Acquisition Function

The AF determines the unknown area to explore and exploit in BO. The higher value of the AF in an unknown area indicates the objective function is optimal or highly possibly optimal. The commonly used AF includes upper confidence bound (UCB), probability of improvement (PI), and expected improvement (EI) [13]. The upper confidence bound is used in this work.

The UCB contains explicit exploitation terms $\mu(x)$ and exploration terms $\sigma(x)$ given by:

$$ucb(x) = \mu(x) + \lambda\sigma(x) \tag{11}$$

where λ is a parameter that balances the trade-off of exploiting and exploring. When λ is small, UCB favors area with higher $\mu(x)$ which could be next to the maximum value of the known sampling points. When λ is large, BO prefers the exploration of the unknown area. However, if λ is too small, BO may obtain a local optimal solution. Corresponding, if λ is too large, BO may focus on exploring the unknown area without procuring the global optimal solution.

Based on multivariate GP and UCB, BO utilizes a set of known sampling to find the new points with maximized AF and approaches to the area with optimal value in each iteration. The complete process of BO has been shown in Fig. 1.

III. Artificial Neural Network and Training Dataset

ANN is a higher-order nonlinear function. Therefore, it has a strong ability to fit the mapping from stack-up parameters to Z_0 and attenuation in the simulation dataset. Multiple-input and multiple-output ANN is easier to train than multiple ADD-GPs [6]. Meanwhile, sufficient dataset and powerful neural network make prediction results more accurate.

A. Structure of Artificial Neural Network

The architecture of the applied ANN model consists of a gated recurrent unit (GRU) [15] and fully connected (FC) layers, as shown in Fig. 2. The applied ANN is more advanced than the ANN with only FC layers, but it is not complex. Less complexity can avoid overfitting and the influence of data noise. The input set $\{x_1, x_2, .., x_n\}$ represents stack-up parameters, as shown in TABLE I and TABLE II.

GRU is a well-known and popular structure of the recurrent neural network (RNN) and is widely used for sequence data processing, such as machine translation and speech recognition. More details of GRU are shown in Fig. 3.

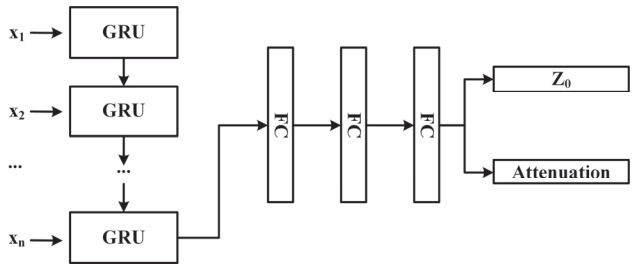


Fig. 2. Structure of artificial neural network.

TABLE I. PARAMETER NUMERICAL RANGES FOR MICROSTRIP LINE DATASET GENERATION

Stack-Up Parameter	Numerical Range	Stack-Up Parameter	Numerical Range
Signal Line Width	40-200um	CL Thickness	32.5-75um
Signal Line Thickness	6, 9, 12, 18um	CCL D_k	2.5-3.5
Gap	80-120um	CCL D_f	0.001-0.01
CL D_k	2.7-3.4	CCL Thickness	25-100um
CL D_f	0.006-0.03		

a.

TABLE II. PARAMETER NUMERICAL RANGES FOR STRIP LINE DATASET GENERATION

Stack-Up Parameter	Numerical Range	Stack-Up Parameter	Numerical Range
Signal Line Width	40-200um	CL Thickness	25-200um
Signal Line Thickness	6, 9, 12, 18um	CCL D_k	2.5-3.5
Gap	80-120um	CCL D_f	0.001-0.01
CL D_k	2.7-4.2	CCL Thickness	25-100um
CL D_f	0.004-0.026		

b.

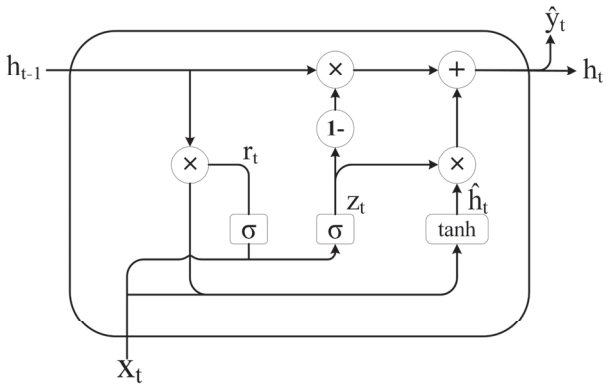


Fig. 3. Gated recurrent unit.

Typically, GRU has an update gate and reset gate. It will process current input and the known information of previous GRU output at the same time. In Fig. 3, h_{t-1} is the output information from the previous GRU block, and x_t is the current input. In this work, each GRU block receives one PCB/package stack-up parameter and combines the current stack-up parameter with known stack-up information. Therefore, h_{t-1} represents known PCB/package stack-up information extracted from previous input stack-up parameters, and x_t is used as the current input stack-up parameter. r_t is reset gate calculated by x_t and h_{t-1}:

$$r_t = \sigma\left(W_r \cdot \left[h_{t-1}, x_t\right]\right) \tag{12}$$

The numeric ranges of the stack-up parameters are different. Hence, it is hard to directly learn the effect and importance of each stack-up parameter from the input numbers. The weights of the reset gate (W_r) will rescale the stack-up parameter values to a comparable range from different numeric ranges. Then, r_t determines how much known information will be combined with current input to generate new information. The new information is presented by $\hat{h}_t$, which is defined as follows:

$$\hat{h}_t = \tanh\cdot\left(W \cdot \left[r_t * h_{t-1}, x_t\right]\right) \tag{13}$$

The update gate (z_t) decides the ratio of new information and known information from the previous GRU block to generate the final output (h_t). z_t and h_t are defined by the following formulas:

$$z_t = \sigma\left(W_z \cdot \left[h_{t-1}, x_t\right]\right) \tag{14}$$

$$h_t = \left(1 - z_t\right) * h_{t-1} + z_t * \hat{h}_t \tag{15}$$

Since different stack-up parameters have different effects on Z_0 and attenuation, and the final result may not be calculated by simply superimposing all effects of stack-up parameters, the gating mechanism of GRU plays an important role. After GRU blocks extract the relationship among stack-up parameters, FC layers finish the final mapping from extracted high-level features to Z_0 and attenuation.

B. Training Data Generation

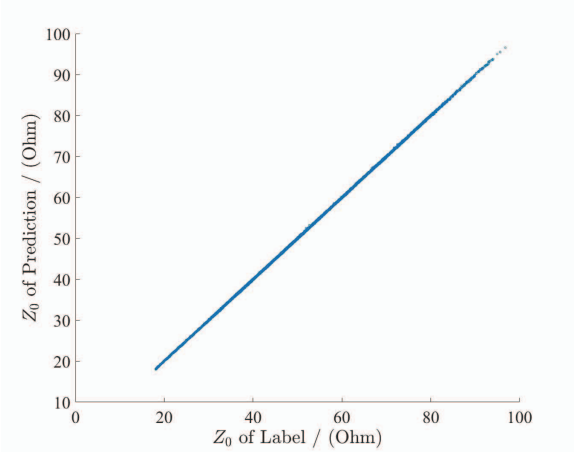
Supervised learning is applied to train the ANN model. Thus, sufficient data are necessary. Ansys Q2D was utilized to generate the dataset. For the microstrip line case and strip line case, two different datasets were generated individually. The microstrip line stack-up is shown in Fig. 6a, and the numerical ranges for every stack-up parameter are listed in TABLE I. In the dataset of the microstrip line, only a 2-dielectric stack-up was considered. In addition, the numerical ranges of stack-up parameters are based on the existing dielectric material library, which makes the dataset meaningful.

Similarly, the dataset for the strip line case was generated, which is shown in TABLE II. To reduce the amount of data, only the 2-dielectric stack-up was considered for data generation. When the BO-ANN algorithm is applied for practical stack-up optimization, the stack-up is extended to multiple-dielectric stack-up by effective dielectric calculation strategy. For this reason, the parameter numerical ranges of CL dielectric are different from the microstrip line case, especially the dielectric thickness. Each dataset includes more than 100,000 stack-ups with corresponding Z_0 and attenuation.

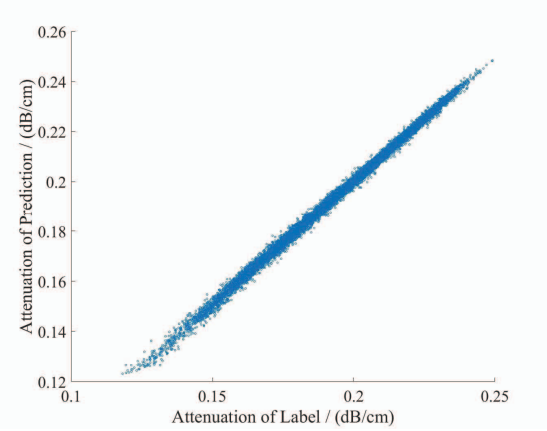
C. Training Results of Artificial Neural Network

The dataset is divided into a training dataset, a testing dataset and a validation dataset. Then, K-fold cross-validation [16] was used during training. After training, the model was tested on the validation dataset. Fig. 4 shows the validation results for the microstrip line at 10 GHz. The ANN model has good performance in Z_0 prediction. As for attenuation prediction, even though the value of attenuation is small, the ANN model can follow its changes.

Likewise, the validation results for the strip line are shown in Fig. 5. The trained model can fit the mapping from stack-up parameters to Z_0 and attenuation well. Meanwhile, the ANN model is 4000 times faster than Ansys Q2D.

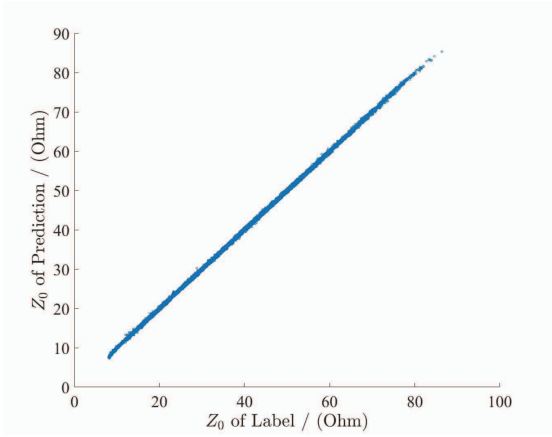


(a) Z_0 prediction result for microstrip line (10 GHz).

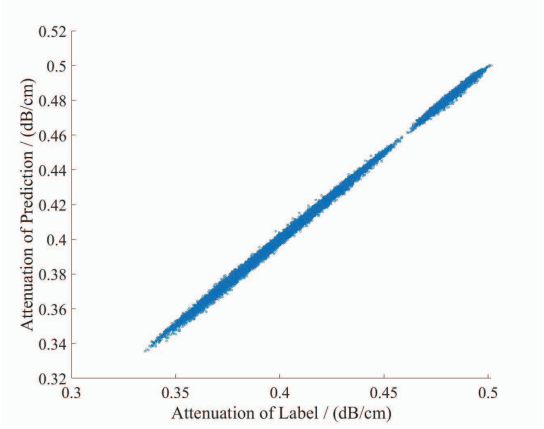


(b) Attenuation prediction result for microstrip line (10 GHz).

Fig. 4. Validation results for microstrip line.



(a) Z_0 prediction result for strip line (10 GHz).



(b) Attenuation prediction result for strip line (10 GHz).

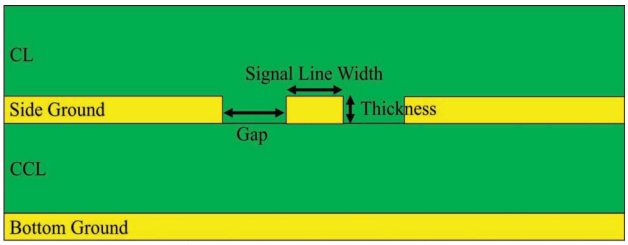
Fig. 5. Validation results for strip line.

IV. STACK-UP OPTIMIZATION

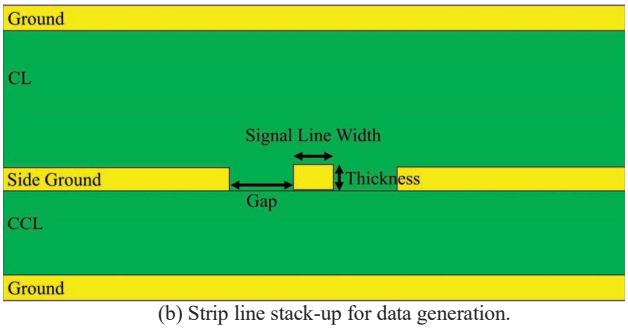
To validate the performance of the proposed algorithm, it was used to optimize practical PCB/package stack-up by choosing dielectric layers from the pre-defined library. The microstrip line case and strip line case are discussed individually.

A. Microstrip Line

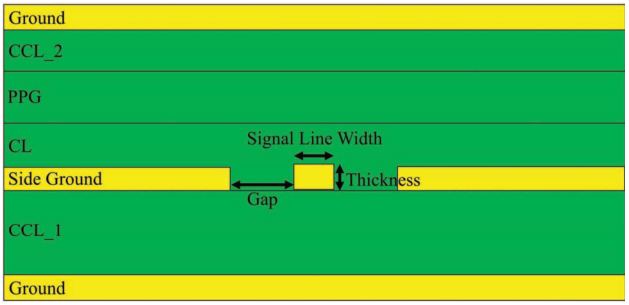
The 2-dielectric stack-up of microstrip line was considered as shown in Fig. 6a. According to the practical PCB/package design, the D_k, D_f and thickness of the specific dielectric is a fixed combination. Thus, BO directly selected dielectrics from the dielectric library instead of individually choosing D_k, D_f, and thickness from a continuous numerical range. Besides, other stack-up parameters was selected from some common numerical ranges as shown in TABLE I, such as signal line width, signal line thickness, and gap. Since it is difficult for BO to find the optimal solution when there only are a few satisfied solutions existing in a large variable space, preprocessing is needed. For example, if there is a severe total height restriction for final stack-up and rare dielectric combinations can satisfy this requirement in the dielectric library, the combinations that do not meet this restriction should be removed from the dielectric library first. Then, BO optimizes the rest stack-up parameters based on the remaining dielectric combinations. In this validation, 50 Ω was set as the target Z_0 and attenuation was minimized as possible. After optimization, Ansys Q2D was used to validate the results. The optimization results for microstrip line are listed in TABLE III. In most cases, the BO



(a) Microstrip line stack-up for data generation and optimization.



(b) Strip line stack-up for data generation.



(c) Strip line stack-up for optimization.

Fig. 6. Stack-up of microstrip line and strip line for data generation and optimization.

could find good solutions according to the predictions of the trained ANN model. But ANN introduced around 1 Ω error in Z_0 prediction, which makes the optimization results somewhat different from the optimal solution. The error of attenuation prediction was smaller than expectation based on training results.

B. Strip Line

In this application, the stack-up was extended to multiple-dielectric layers as shown in Fig. 6c, which is also common in practical designs. Since the training data of the ANN model only include a 2-dielectric stack-up, the BO-ANN algorithm firstly selected parameters according to the stack-up shown in Fig. 6b. Unlike the microstrip line case, BO only selected CCL from the given dielectric library while choosing CL D_k, CL D_f, and CL thickness from specific numerical ranges. For the extension from 2-dielectric stack-up to multiple-dielectric stack-up, a method of approximate effective dielectric calculation is defined as follows:

$$T_{eff} = T_{CL} + T_{PPG} + T_{CCL}$$

$$D_{k\_eff} = \frac{T_{CL} * D_{k\_CL}}{T_{eff}} + \frac{T_{PPG} * D_{k\_PPG}}{T_{eff}} + \frac{T_{CCL} * D_{k\_CCL}}{T_{eff}}$$

$$D_{f\_eff} = \frac{T_{CL} * D_{f\_CL}}{T_{eff}} + \frac{T_{PPG} * D_{f\_PPG}}{T_{eff}} + \frac{T_{CCL} * D_{f\_CCL}}{T_{eff}} \quad (16)$$

where T presents thickness. After calculating all dielectric combinations from the given dielectric library, an effective dielectric list was obtained. Then, the effective dielectric list and CL parameters selected by the BO-ANN algorithm were used to calculate scores for every dielectric combination. The scores calculation formula is given as follows:

$$Scores = \frac{abs\left(T_{eff} - T_{BO}\right)}{3} + 3 * abs\left(D_{k\_eff} - D_{k\_BO}\right) + 100 * abs\left(D_{f\_eff} - D_{f\_BO}\right) \quad (17)$$

where T_{eff}, D_{k_eff} and D_{f_eff} are calculated in (16). T_{BO}, D_{k_BO} and D_{f_BO} are CL thickness, CL D_k and CL D_f selected by BO-ANN algorithm.

Finally, the dielectric combinations with the top 5 scores were chosen to form a multiple-dielectric stack-up for the strip line. The optimization result of the proposed BO-ANN algorithm with an effective dielectric strategy is listed in TABLE IV. Also, the Ansys Q2D was utilized to verify the performance of the proposed algorithm. According to the predictions of ANN, the BO-ANN algorithm found the proper stack-ups with Z_0 and attenuation close to targets. In comparison with Q2D results, there are still some errors. On the one hand, the prediction error of the trained ANN Model cased the difference between the proposed algorithm and Q2D. On the other hand, the effective dielectric calculation is only a rough estimate that introduces another error.

TABLE III. OPTIMIZATION RESULTS FOR MICROSTRIP LINE

Optimization Result	#1	#2	#3	#4	#5
Signal Line Width (um)	100	91	132	130	94
Signal Line Thickness (um)	6	12	18	18	12
Gap (um)	80	82	102	108	113
CCL D_k	3.33	3.29	3.20	3.20	3.29
CCL D_f	0.0039	0.0027	0.005	0.005	0.0027
CCL Thickness (um)	50	50	75	75	50
CL D_k	3.25	2.7	3.10	3.90	2.70
CL D_f	0.019	0.004	0.006	0.005	0.004
CL Thickness (um)	17.5	32.5	27.5	50	32.5
Z_0 Predicted by ANN (Ω)	49.738	49.846	50.875	50.359	50.027
Attenuation Predicted by ANN (dB/cm)	0.3447	0.295	0.2383	0.2412	0.2880
Z_0 Validated by Q2D (Ω)	48.749	49.192	49.511	49.592	49.501
Attenuation Validated by Q2D (dB/cm)	0.35	0.296	0.2402	0.2509	0.2911

c.

TABLE IV. OPTIMIZATION RESULTS FOR STRIP LINE

Optimization Result	#1
Signal Line Width (um)	68
Signal Line Thickness (um)	12
Gap (um)	89
CCL_1 D_k	3.33
CCL_1 D_f	0.0037
CCL_1 Thickness (um)	50
CL D_k	3.19
CL D_f	0.0210
CL Thickness (um)	27.5
PPG D_k	3.70
PPG D_f	0.0160
PPG Thickness (um)	35
CCL_2 D_k	3.20
CCL_2 D_f	0.0050
CCL_2 Thickness (um)	75
Z_0 Predicted by ANN (Ω)	49.985
Attenuation Predicted by ANN (dB/cm)	0.3715
Z_0 Validated by Q2D (Ω)	49.949
Attenuation Validated by Q2D (dB/cm)	0.4627

Optimization Result	#2	#3	#4	#5
Signal Line Width (um)	78	80	80	68
Signal Line Thickness (um)	9	6	6	12
Gap (um)	112	115	115	89
CCL_1 D_k	3.30	3.33	3.33	3.33
CCL_1 D_f	0.0039	0.0037	0.0037	0.0037
CCL_1 Thickness (um)	50	50	50	50
CL D_k	3.00	2.90	3.00	3.10
CL D_f	0.0050	0.0050	0.0050	0.0060
CL Thickness (um)	75	50	75	27.5
PPG D_k	3.20	3.32	3.20	3.34
PPG D_f	0.0040	0.0055	0.0040	0.0320
PPG Thickness (um)	45	55	45	35
CCL_2 D_k	3.40	3.48	3.40	3.20
CCL_2 D_f	0.0020	0.0050	0.0043	0.0050
CCL_2 Thickness (um)	7	25	12	75
Z_0 Predicted by ANN (Ω)	49.831	50.136	49.785	50.06
Attenuation Predicted by ANN (dB/cm)	0.3482	0.3477	0.3512	0.3861
Z_0 Validated by Q2D (Ω)	49.115	49.123	49.110	50.223
Attenuation Validated by Q2D (dB/cm)	0.3459	0.3614	0.3656	0.4355

d.

V. CONCLUSION

This paper proposes the BO-ANN algorithm to implement fast PCB/package stack-up optimization in terms of both Z_0 and attenuation. The effective dielectric calculation strategy endows the BO-ANN algorithm with good generalization ability, making the proposed algorithm applicable to more complex applications. For practical microstrip line and strip line stack-up optimizations, the difference between the Z_0 of the optimized stack-up and target Z_0 was less than 1Ω when target Z_0 is set to 50 Ω.

REFERENCES

[1] I. Erdin and R. Achar, "Multipin optimization method for placement of decoupling capacitors using a genetic algorithm," *IEEE Transactions on Electromagnetic Compatibility*, vol. 60, no. 6, pp. 1662-1669, Dec. 2018.

[2] J. Juang, L. Zhang, Z. Kiguradze, B. Pu, S. Jin and C. Hwang, "A modified genetic algorithm for the selection of decoupling capacitors in PDN design," *2021 IEEE International Joint EMC/SI/PI and EMC Europe Symposium*, Raleigh, NC, USA, 2021, pp. 712-717.

[3] L. Zhang, W. Huang, J. Juang, H. Lin, B. -C. Tseng and C. Hwang, "An enhanced deep reinforcement learning algorithm for decoupling capacitor selection in power distribution network design," *2020 IEEE International Symposium on Electromagnetic Compatibility & Signal/Power Integrity (EMCSI)*, Reno, NV, USA, 2020, pp. 245-250.

[4] C. E. Rasmussen, and C. K. I. Williams, Gaussian Processes for Machine Learning. MIT Press. Cambridge, Massachusetts, 2006.

[5] K. Scharff, H. M. Torun, C. Yang, M. Swaminathan and C. Schuster, "Bayesian optimization for signal transmission including crosstalk in a via array," *2020 International Symposium on Electromagnetic Compatibility - EMC EUROPE*, Rome, Italy, 2020, pp. 1-6.

[6] H. M. Torun, M. Larbi and M. Swaminathan, "A Bayesian framework for optimizing interconnects in high-speed channels," *2018 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO)*, Reykjavik, Iceland, 2018, pp. 1-4.

[7] Z. Kiguradze, J. He, B. Mutnury, A. Chada and J. Drewniak, "Bayesian optimization for stack-up design," *2019 IEEE International Symposium on Electromagnetic Compatibility, Signal & Power Integrity (EMC+SIPI)*, New Orleans, LA, USA, 2019, pp. 629-634.

[8] D. K. Duvenaud, H. Nickisch, and C. E. Rasmussen, "Additive gaussian processes," in *Advances in neural information processing systems*, 2011

[9] J. Mockus, "On Bayesian methods for seeking the extremum," Optimization Techniques, 1974, pp. 400-404.

[10] J. Mockus, "On Bayesian methods for seeking the extremum and their application," IFIP Congress, 1977, pp. 195-200.

[11] J. Mockus, Bayesian Approach to Global Optimization. Kluwer Academic Publishers, Dordrecht, 1989.

[12] McPherson G. Statistics in scientific investigation: its basis, application, and interpretation. Springer Science & Business Media, 2013.

[13] K. Il. Park, Fundamentals of Probability and Stochastic Processes with Applications to Communications, Springer, 2018.

[14] F. Archetti, A. Candelieri, Bayesian optimization and data science. Springer International Publishing, 2019

[15] Cho K, Van Merriënboer B, Bahdanau D, et al. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259, 2014.

[16] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT press, 2016.