2020, 37th NATIONAL RADIO SCIENCE CONFERENCE
(NRSC 2020), September 8-10, 2020

German University in Cairo (GUC), Cairo, Egypt

APPLICATION OF ARTIFICIAL NEURAL NETWORKS TO THE AUTOMATION OF BANDGAP REFERENCE SYNTHESIS

Nabil R. Soliman, Karim D. Khalil, Ahmed M. Abd El Khalik, and Hesham Omran
The Integrated Circuits Laboratory, Electronics and Communications Engineering Department, Faculty of Engineering, Ain Shams University, Cairo, Egypt. Email: hesham.omran@eng.asu.edu.eg

This work was supported by the Information Technology Industry Development Agency (ITIDA) under Grant PRP2018.R24.7.

ABSTRACT

Bandgap voltage references are present in virtually every analog/mixed-signal system. However, their design still remains a time-consuming procedure that requires extensive designer expertise and validation. In this paper, an automated bandgap synthesis procedure is used to generate a dataset that maps the specifications of the synthesized bandgap reference circuit to their corresponding designer's degrees of freedom. This dataset is then used to train a neural network to predict the choice of the degrees of freedom in order to meet arbitrary circuit specifications specified by the user including variations due to design corners and random mismatch. The automated bandgap synthesis procedure uses precomputed look-up tables rather than invoking a circuit simulator in the loop, which enables generating a large dataset of training examples in short time. The choice of the degrees of freedom predicted by the neural network is then re-fed to the bandgap synthesis procedure to verify the accuracy of the prediction and obtain the complete solution of the synthesized circuit. The results demonstrate that the trained neural network is capable of making successful predictions of good accuracy in a wide multi-dimensional design space.

Keywords: *Bandgap voltage reference; artificial neural networks; machine learning; analog design automation.*

I. INTRODUCTION

Because we live in an analog world, analog integrated circuits will always remain an indispensable part in electronic systems [1]. The tremendous growth in the field of electronic design automation has substantially facilitated the design workflow of digital integrated circuits. However, the automation of analog integrated circuits still remains a considerable challenge, mainly due to the complex multi-objective nature of the analog design problem and the sensitive interdependence between analog circuit parameters. Consequently, the greater part of the task of analog design is completely left up to the designer's expertise and knowledge.

Such a gap in the extent of automation between analog and digital design flows has caused analog design to become the bottleneck of mixed-signal chips development cycles, in spite of the analog building blocks consisting of a much smaller number of devices compared to their digital counterparts. With the increasing requirements on the specifications and development times of analog blocks, analog design automation has become a topic of significant economic and technical importance [2].

The task of analog design automation can roughly be divided into three separate problems: the automation of the topology selection, the automation of the device sizing, and the automation of the layout generation [3]. In the last few decades, different approaches were applied to the problem of device sizing, the most popular of which is the simulation-based optimization approach [4], [5]. This approach relies on invoking a simulator in the loop in an iterative procedure, which results in very long run-times, especially if several degrees of freedom exist. A relatively new and interesting approach, which only recently became feasible due to the increase in the available computing power, is the application of machine learning and neural networks to the task of analog circuit sizing [6]. A notable issue that must nonetheless be considered is the difficulty of large training set generation, owing to the need to invoke the simulator several times to generate every training example. A promising design methodology that could resolve this problem is the use of precomputed lookup tables (LUTs) [7]-[10]. This methodology was recently used to automatically synthesize and characterize a bandgap reference voltage reference circuit given designer's degrees of freedom (sizing and bias parameters), without invoking the simulator in the loop [11]. However, what the designer is actually interested in is what degrees of freedom shall he use in order to achieve a given set of specifications. Nevertheless, this important requirement is not addressed by the synthesis procedure proposed in [11]. In this paper, we address this requirement by using a dataset generated using the automated bandgap synthesis procedure in [11] to train an artificial neural network. The neural network is then used to generate the sizing and bias parameters of a

978-1-7281-6819-7/20/$31.00 ©2020 IEEE

bandgap voltage reference based on input specifications as shown in Fig. 1. The training set is generated using a vectorized LUT-based procedure, which eliminates the need to invoke the simulator in the loop and enables the generation of a large number of training examples in short time.

The rest of the paper is organized as follows. Section II discusses the generation of the dataset and the bandgap to neural network interface. Section III describes the training of the neural network. Section IV presents the obtained results, and Section V presents the conclusion.

II. Bandgap Synthesis Procedure to Neural Network Interface

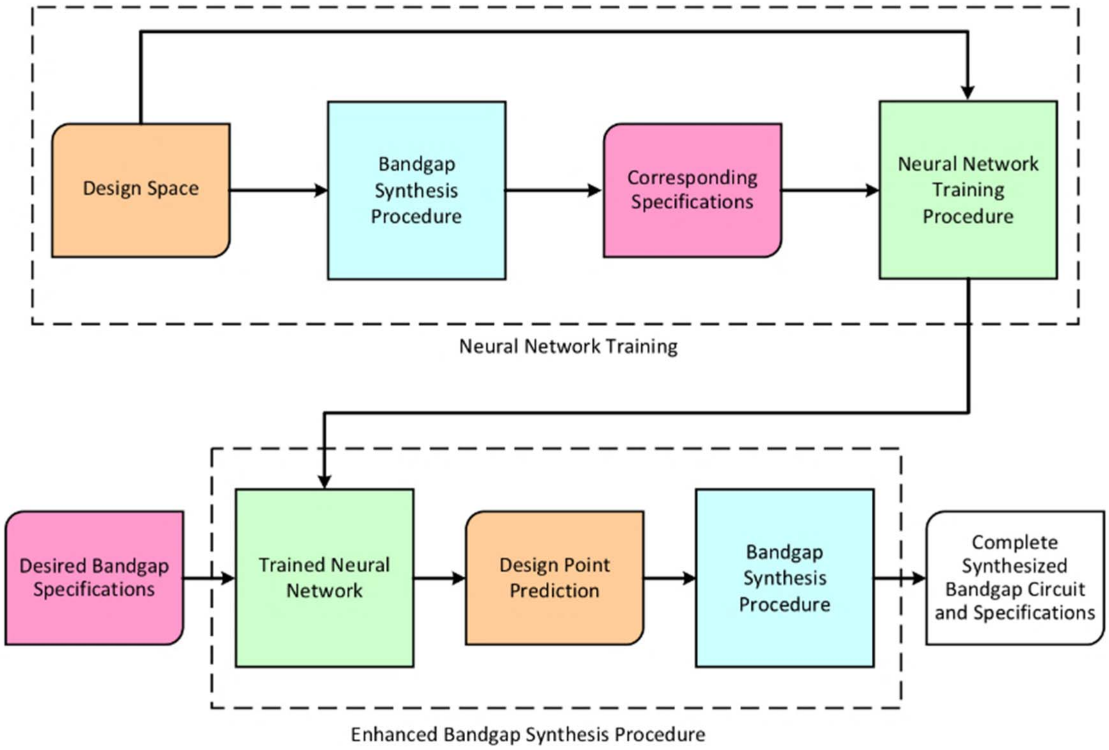

Design Space
Bandgap Synthesis Procedure
Corresponding Specifications
Neural Network Training Procedure
Neural Network Training
Desired Bandgap Specifications
Trained Neural Network
Design Point Prediction
Bandgap Synthesis Procedure
Complete Synthesized Bandgap Circuit and Specifications
Enhanced Bandgap Synthesis Procedure

Fig. 1: Block diagram of the bandgap synthesis procedure to neural network interface.

The bandgap synthesis procedure in [11] uses the LUTs of MOSFET and BJT devices for a desired technology node, in addition to the choice of degrees of freedom (gm/ID) and length of PMOS and NMOS devices, and total bias current) to synthesize the bandgap reference circuit shown in Fig. 2.

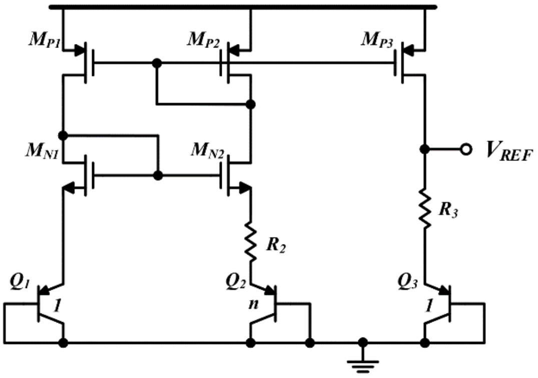


Fig. 2: Circuit diagram of the synthesized bandgap reference circuit.

The procedure calculates the device widths and the values of the resistors (R2 and R3) to precisely cancel the first-order temperature dependence of the output reference voltage. It then computes the bandgap circuit specifications (power supply rejection, noise density, percentage of output variation due to random mismatch, percentage of output variation due to PVT corners, and area) of the synthesized circuit, all without invoking the simulator. This allows the designer to rapidly explore the circuit design space and the behaviour of key performance metrics.

Using a vectorized implementation of the aforementioned bandgap synthesis procedure, it becomes possible to quickly generate large datasets that map the design space (the space of the designer's degrees of freedom) to circuit specifications. A 5-dimensional design space consisting of different channel length and gm/ID for PMOS and NMOS devices in addition to total bias current was input to the bandgap synthesis procedure to generate a set of specifications corresponding to each design point in the design space. The design space was generated such that the length of PMOS and NMOS devices was swept from 1 µm to 8 µm, the gm/ID from 10 to 20 S/A, and the total bias current from 10 µA to 100 µA, with each sweep consisting of 8 linearly spaced points, thus generating a dataset with 32768 unique bandgap reference circuits.

A neural network was trained using the generated dataset to learn to predict the design points that satisfy arbitrary input specifications as shown in Fig. 1. The neural network's predictions are then input once more to the bandgap synthesis procedure in order to obtain a complete description of the circuit's bias and device sizing and estimate the error of the specifications corresponding to the design point predicted by the neural network versus the input specifications. The interface created between the bandgap synthesis procedure and the neural network effectively enhances the work proposed in [11], allowing the designer to synthesize a precision bandgap reference circuit based on the desired specifications, rather than an arbitrary choice of design point.

III. Neural Network Model Selection and Training Procedure

The input features of the neural network (the bandgap specifications) were scaled to have a minimum and maximum values of -1 and 1, respectively. The MATLAB Deep Learning Toolbox was used to train the neural network using Levenberg-Marquardt algorithm. Rectified linear unit (ReLU) function was chosen as the activation function for the hidden layers, with the output layer having a linear activation function. The cost function is the Mean Squared Error (MSE), which, for the predictions made by the neural network on a specific dataset, is given by:

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(Y'_{(i)} - Y_{(i)})^T(Y'_{(i)} - Y_{(i)}) \qquad (1)$$

where $Y'_{(i)}$ and $Y_{(i)}$ represent the outputs predicted by the neural network and the outputs of the dataset used to train the neural network for the i^{th} training example, respectively, and m represents the number of examples in the dataset. Note that the cost function regularization term was not used in (1) as the network did not suffer from high variance.

Two additional datasets (2000 points each) were generated for cross-validation (CV) and testing. The training and CV MSE were plotted for different neural network models (different number of hidden layers and number of units per layer) as shown in Fig. 3 and Fig. 4 to select the model that has the best trade-off between bias and variance. It can be noted that the model does not yet suffer from overfitting even at the largest tried network. Thus, a 50-50-50 neural network model was selected, which also has the advantage of short training time.

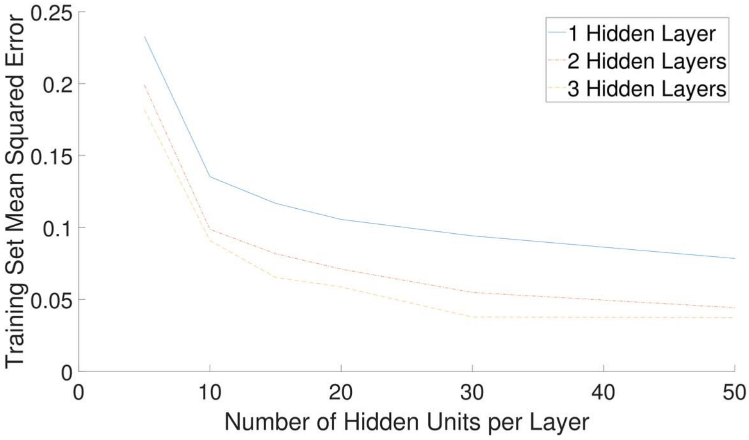


Fig. 3: Training set mean squared error for different number of hidden layers and units in the neural network.

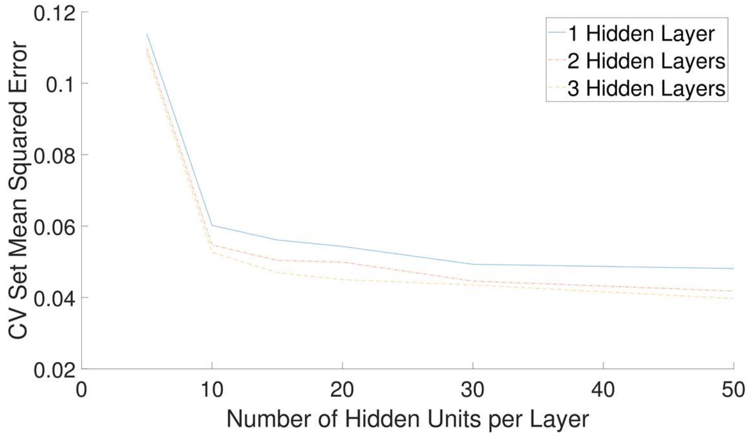


Fig. 4: Cross-validation set mean squared error for different number of hidden layers and units in the neural network.

The learning curves were then plotted for the selected model as shown in Fig. 5 to verify the sufficiency of the dataset and ensure that the model generalizes well to new data. The selected neural network model had a training MSE of 0.037 and a CV MSE of 0.04. The learning curves show that the chosen model indeed generalizes well (the training and cross-validation errors are very close) and the training set size is satisfactory.

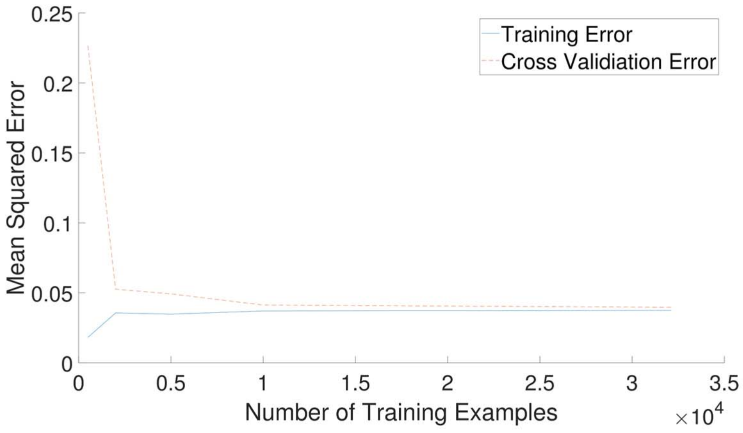


Fig. 5: Learning curves for the selected neural network model.

IV. Results and Discussion

The bandgap synthesis procedure enhanced by the trained neural network was used as shown in Fig. 1 to synthesize bandgap reference circuits for different sets of specifications. In order to measure the accuracy of the synthesis procedure, the neural network was used to synthesize 2,000 unique bandgap circuits. Next, the average errors between the specifications required by the user (the input specifications) and the actual metrics of the circuit synthesized by the neural network are shown in Fig. 6. It can be noted that the errors in power supply rejection (PSR) and percentage corner variation are quite low. The errors in noise, mismatch variation, and area are relatively higher but still tolerable.

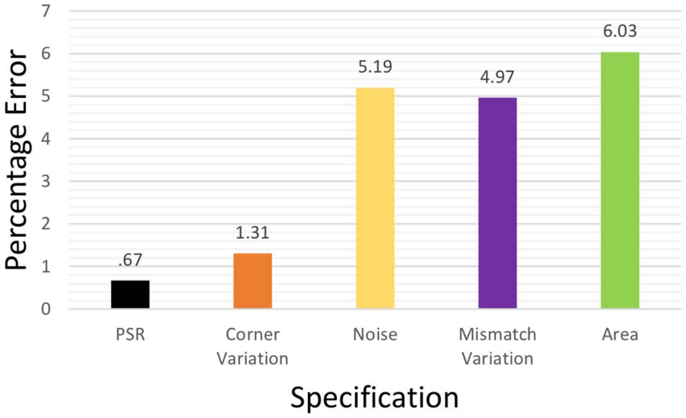
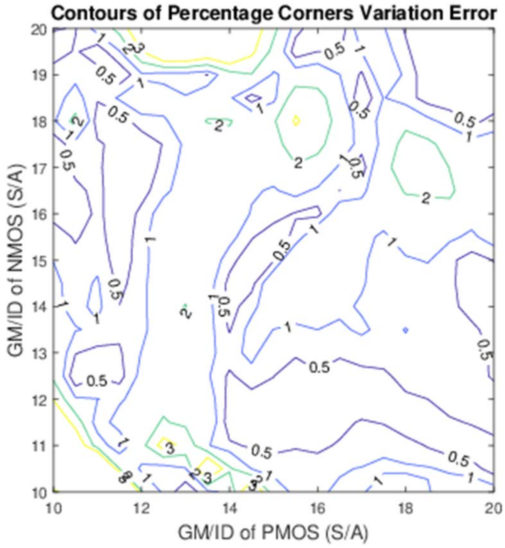
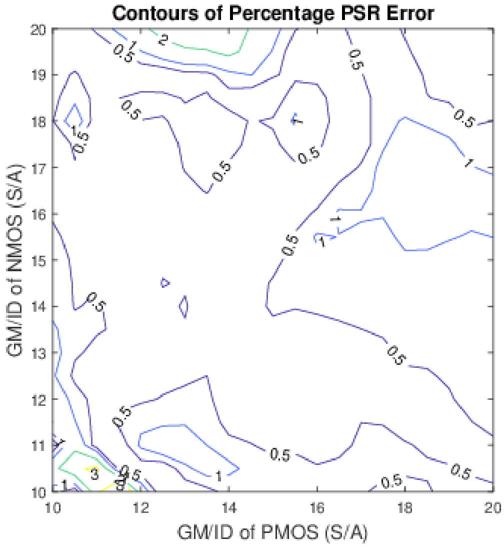


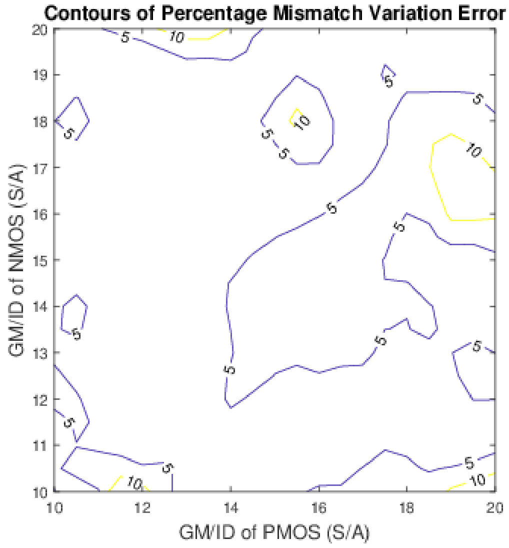
Fig. 6: Specifications percentage error averaged over the entire test set.

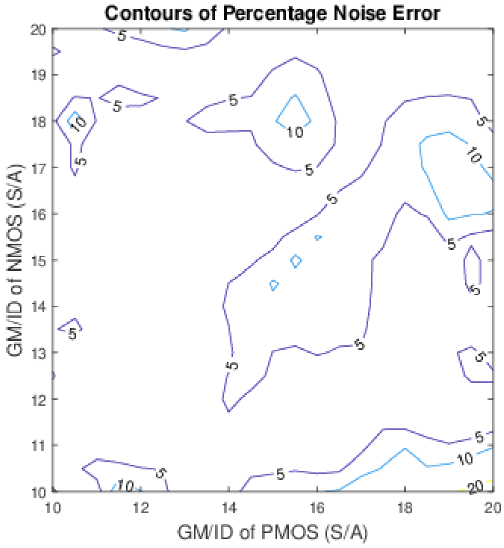
In order to further investigate the accuracy of the trained network's predictions, we generated additional test sets in which only two degrees of freedom are swept at a time to generate the input specifications. The contours of the error of each specification against the specific degrees of freedom are plotted in Fig. 7 and Fig. 8. It can be seen that the errors are tolerable over this wide design space. The percent of error relatively increases when the design uses a short device length, or when there is a great disparity between the gm/ID of the PMOS and NMOS devices. That is to say, the predictions made by the neural network are most error prone near the extremes of the design space used to train it. Consequently, adding margins to the design space used to generate the training examples can guarantee accurate predictions by the neural network.

Table 1 shows an example of a design synthesised by the trained neural network to realize a given set of input specifications. The table compares the user's input (required) specifications, the specifications of the circuit sized by the neural network as calculated by the bandgap synthesis procedure, and the simulation results of the synthesized circuit using Cadence Spectre. The results show that the required specifications are achieved with reasonable accuracy. In addition, there is a good agreement between the synthesized and the simulated results.


Contours of Percentage Corners Variation Error
GM/ID of NMOS (S/A)
GM/ID of PMOS (S/A)
10
12
14
16
18
20
0.5
1
2
3


Contours of Percentage PSR Error
GM/ID of NMOS (S/A)
GM/ID of PMOS (S/A)
10
12
14
16
18
20
0.5
1
2
3


Contours of Percentage Mismatch Variation Error
GM/ID of NMOS (S/A)
GM/ID of PMOS (S/A)
10
12
14
16
18
20
5
10


Contours of Percentage Noise Error
GM/ID of NMOS (S/A)
GM/ID of PMOS (S/A)
10
12
14
16
18
20
5
10
20

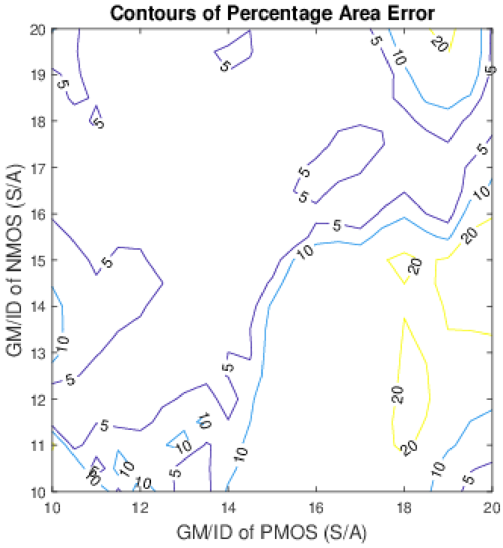
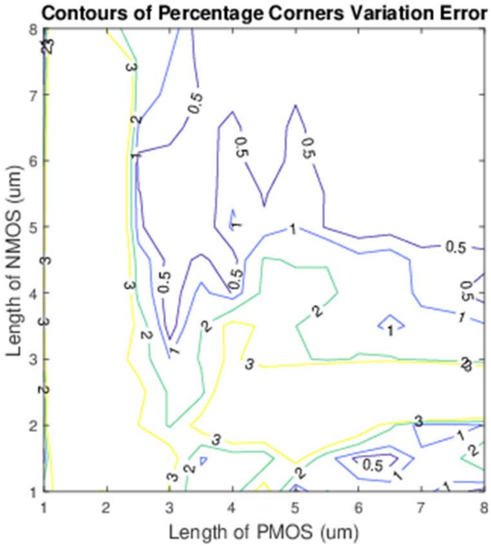
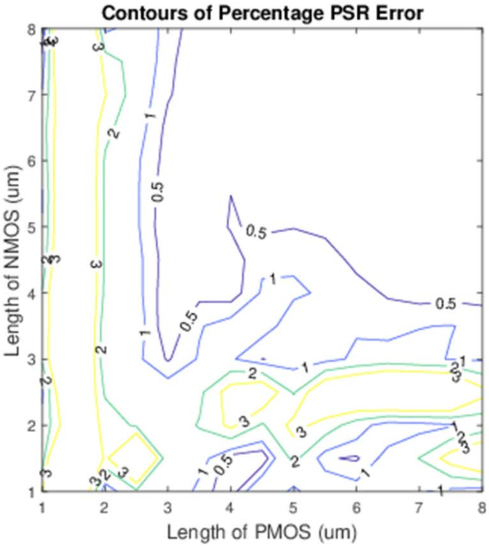
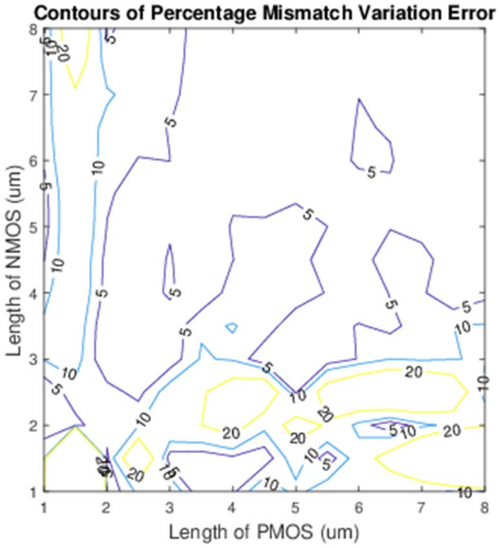
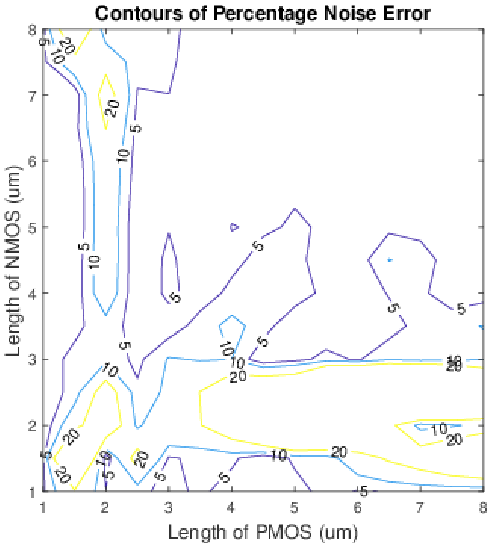

Contours of Percentage Area Error
GM/ID of NMOS (S/A)
GM/ID of PMOS (S/A)
10
12
14
16
18
20
5
10
20

Fig. 7: Contours of percentage error in output specifications versus gm/ID.


Contours of Percentage Corners Variation Error


Contours of Percentage PSR Error


Contours of Percentage Mismatch Variation Error


Contours of Percentage Noise Error

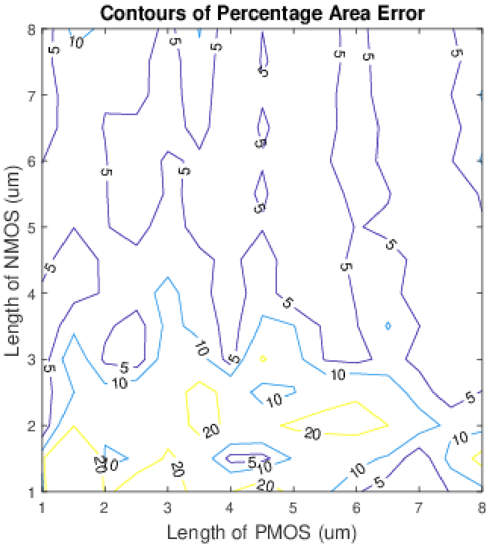

Contours of Percentage Area Error

Fig. 8: Contours of percentage error in output specifications versus device length.

Table 1: Comparison Between the Input, Synthesized, and Simulated Specifications for the Complete Enhanced Bandgap Synthesis Procedure

Specification	***Input***	***Synthesized***	***Simulated***
PSRR	-30.27dB	-30.39dB	-29.5dB
% Corner Variation	2%	1.975%	1.96%
Noise Density	5.6µV/sqrt(Hz)	5.9 µV/sqrt(Hz)	6 µV/sqrt(Hz)
% Mismatch Variation	0.85%	0.9%	0.94%
Area	11800µm^2	13680µm^2	13680µm^2

V. Conclusion

The usage of neural network in designing a complete bandgap reference circuit given a set of specifications was explored. A training dataset that covers the whole design space was generated using a vectorized time efficient bandgap synthesis algorithm. A cross-validation (CV) dataset was then used to compare various neural networks with different architectures to decide on the best architecture. Learning curves showing training and CV errors versus the training set size were plotted to ensure a good trade-off between bias and variance. The trained neural network was tested, and the obtained results were then returned to the bandgap synthesis algorithm to ensure that the specifications were met. The results show that neural networks, when used in tandem with systematic design algorithms, can create new analog design automation solutions of reasonable accuracy that can synthesize precision bandgap circuits, and can be generalized to other analog building blocks.

References

[1] B. Razavi, *Design of analog CMOS integrated circuits*, 2nd ed., McGraw-Hill Education, 2017.

[2] A. C. Kammara, L. Palanichamy, and A. König, “Multi-objective optimization and visualization for analog design automation,”, *Complex & Intelligent Systems*, vol. 2, no. 4, pp. 251-267, Dec 2016.

[3] M. Barros, J. Guilherme, and N. Horta, “State-of-the-art on analog design automation,” *Studies in Computational Intelligence*, vol. 294, Jan. 2010.

[4] Virtuoso Analog Design Environment GXL. Cadence Design Systems. [Online]. Available: https://www.cadence.com [Accessed Aug. 31, 2019]

[5] N. Lourenco, R. Martins, A. Canelas, R. Povoa, N. Horta, and E. Moutaye, “Hard and soft constraints for multi-objective analog IC sizing optimization,” in *2019 16th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD),* July 2019, pp. 285–288.

[6] N. Lourenco, E. Afacan, R. Martins, F. Passos, A. Canelas, R. Povoa, N. Horta, and G. Dundar, “Using polynomial regression and artificial neural networks for reusable analog ic sizing,” in *2019 16th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, July 2019, pp. 13–16.

[7] P. Jespers and B. Murmann, *Systematic Design of Analog CMOS Circuits Using Pre-Computed Lookup Tables.* Cambridge University Press, 2017.

[8] M. N. Sabry, H. Omran, and M. Dessouky, “Systematic design and optimization of operational transconductance amplifier using gm/ID design methodology,” *Microelectronics Journal*, vol. 75, pp. 87 – 96, 2018.

[9] O. A. Abu-El-ela, A. A. Aldaly, Y. Abd-El-azim, and H. A. Omran, “Design automation of a fully differential switched capacitor amplifier using a telescopic cascode OTA,” in *2018 35th IEEE National Radio Science Conference (NRSC)*, March 2018, pp. 385–392.

[10] A. M. Abdel-Aziz, O. H. Fawzy, H. O. Zahran, K. A. Tera, M. N. Sabry, and H. Omran, "Design automation of fully-differential folded cascode OTA using nested iterations," in *2018 International Japan-Africa Conference on Electronics, Communications and Computations (JAC-ECC)*, Dec 2018, pp. 143–146.

[11] H. Omran, M. H. Amer, and A. M. Mansour, "Systematic design of bandgap voltage reference using precomputed lookup tables," *IEEE Access*, vol. 7, pp. 100 131–100 142, 2019.