

Automated Design of Analog Circuits using Machine Learning Techniques

Devi S

devis@ee.iitb.ac.in

Gourav Tilwankar

gouravtil@ee.iitb.ac.in

Rajesh Zele

rajeshzele@ee.iitb.ac.in

Department of Electrical Engineering, IIT Bombay, Mumbai, India

Abstract—This work presents methodology for an automated design of analog circuits using global Artificial Neural Network (ANN) for an optimised dataset. The optimised dataset is generated using simulation based g_m/I_d technique, which reduces the dataset size and also the time required for data collection and analysis. Automated analog circuit design is implemented using ANN based supervised learning technique for a common source amplifier and a two stage single-ended opamp. The results obtained are compared with unsupervised (Reinforcement Learning algorithm) and supervised learning technique (Genetic Algorithm based local ANN). The comparison results shows that the proposed g_m/I_d technique based ANN model gives a better accuracy in terms of score and mean square error (MSE).

Index Terms—Machine Learning, Reinforcement learning (RL), Artificial Neural Networks (ANN), Genetic Algorithm (GA), Python

I. INTRODUCTION

Analog circuits play an important role in processing real world signals and interfacing them with digital signal processing units. The design of analog circuits is challenging as the parameters of the analog circuits are interrelated and difficult to tune to achieve the desired performance. Optimizing one parameter can result in the degradation of another. Designers are always trying to achieve an optimum trade-off between the performance metrics. Analog circuits have non-linearities associated with their components. Hence, the final optimum design of a circuit takes a large number of iterations for the fine-tuning of the circuit parameters. The previous works in the automated design of analog circuits starts with a particular topology and tries to optimize the parameters such as transistor width and bias current based on some cost function, which further depends on the performance metrics. The two main optimization approaches are equation-based and simulation-based design. Equation based approach utilizes analytical design equations to optimize the performance metrics. These equations have to go through a large design space to reach an optimum solution. The accuracy of these solutions will be limited as the equations considered for the optimization will not account for the device non-linearities. The simulation-based approach works in conjunction with the SPICE simulations, to reach the global optimum solution. However, this process requires high computational cost due to large number of SPICE runs. There are various other techniques that make use of nonlinear data space exploration like GA, simulated annealing, and particle swarm intelligence. The simulated annealing method and the swarm-based method sometimes gets stuck at a local optimum solution rather than finding a global optimum solution requiring a longer time to converge. In the case of GA, a large population is required to achieve

a reasonable solution requiring large SPICE simulations. The advantage of the GA-based optimization technique is that the population used for evaluations are independent of each other enabling parallel computations. Some of these above-said algorithms work along with local minimum search techniques to reach the optimum solution.

In this paper, the different techniques of automated design of analog circuits using Machine Learning (ML) methods are analyzed. We propose a new g_m/I_d method of data collection which reduces the overall time for analysis. The previous work done in [1] uses ANN which can determine a functional mapping with a certain precision using a dataset. The time required for the evaluation of the model and to attain the design specifications is much less than the equation-based or the simulation-based methods. The ANN model provides a global optimum solution, requiring a huge dataset resulting in a large number of SPICE simulations. The paper is organized as follows: Section II describes the unsupervised learning technique using reinforcement learning. Forward propagation and backward propagation techniques using supervised learning method are given in Section III. Proposed model using global ANN with optimized dataset is detailed in Section IV followed by conclusions in Section V.

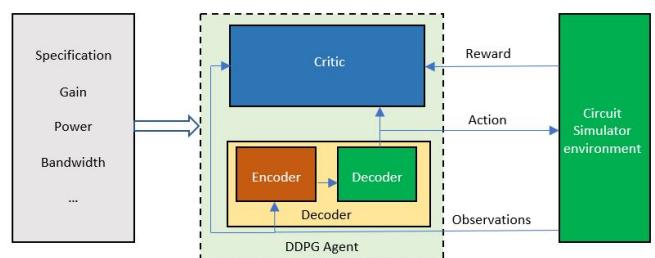


Fig. 1. L2DC Method [2]

II. UNSUPERVISED LEARNING TECHNIQUE

The automated design of analog circuits using ML techniques is introduced in [2]. It explains the method L2DC (Learning to Design Circuits), which uses the RL algorithm for the design of circuit parameters as shown in Figure 1. In this work, the author have demonstrated the simulation results for trans-impedance amplifiers.

The technique starts to train the RL agents without any specific rules or constraints. In each step, the agent obtains observation from the environment, producing an action for the environment (values for design parameters), and then determines the reward as a function of gain, power, bandwidth, area, etc. The observations from the environment include

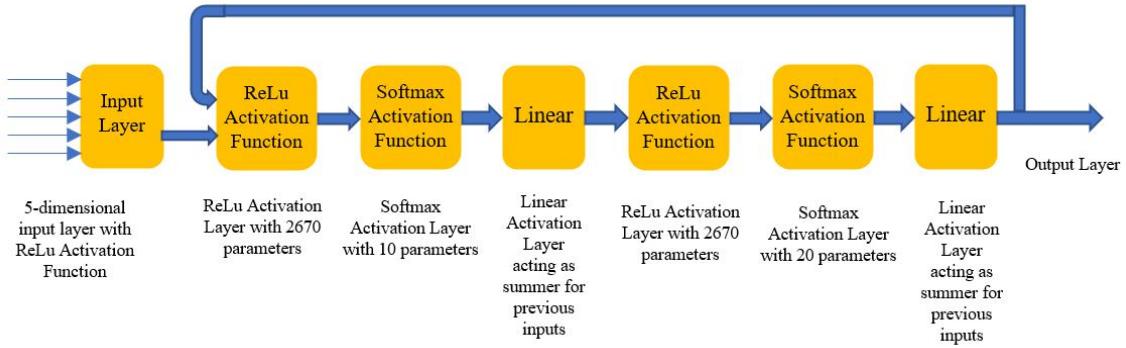


Fig. 2. Flowchart of Machine Learning Model

DC operating points, AC magnitude, and phase response which are obtained from the SPICE simulator. The reward is given in terms of the figure of merit (FoM). The optimum circuit parameters for particular design specifications can be achieved by maximizing the reward. As shown in Figure 1, the method starts with certain hard design constraints and optimizing targets. We will have a x, y where x and y are vectors, x defines the parameters of the components and y denote the design constraints. In the RL algorithm the agent is trying to find $f: x \rightarrow y$ by maximizing the reward which is defined as the ratio $q_c(x) = f_c(x)/y_c$, if performance metric larger than specification, otherwise $q_c(x) = y_c/f_c(x)$. A Deep Deterministic Policy Gradient (DDPG) agent which uses an encoder-decoder network is also employed for the conversion of observations to actions. The steps in the DDPG can be broken into four steps, namely experience replay, actor and critic network updates, target network updates, and exploration. In each iteration, the four states, state, action, reward, and next state are stored in the replay buffer. The mini-batches out of it are sampled to update the reward. With the help of the experience replay, the uncorrelated data can be generated for running each iteration. The actor and value network will compute the current value and find the MSE from the actual value and attempting to minimize this MSE. The critic network is working on the basis of the multi-layer perceptron technique. Once the MSE is minimized, the target network values will be reached and soft updating is done at this stage. In RL for discrete action spaces, a study is done by probabilistically picking an arbitrary action such as epsilon greedy or Boltzmann exploration [7]. Although the amount of dataset required is reduced compared to supervised learning technique the time required for reaching at the optimum solution based on bandwidth and gain at the expense of power remains same as the simulation tools are also involved in the each iteration for calculating the reward.

III. SUPERVISED LEARNING TECHNIQUE

A. Forward Propagation Technique

The supervised learning techniques requires us to train the model on some data points that includes both input and output values. Figure 2 shows the algorithm based on forward propagation technique implemented for creating Machine Learning Model coded in Python Language. Initially the input layer takes circuit parameters as input, calculates the first layer output and after series of steps gives the final output of first

iteration. This output is then fed back to one of the hidden layer which then works on minimizing the error between two outputs. This is done by adjustment of weights and bias values of hidden layers. Finally, the output specifications are achieved after a stated number of iterations/loops. As an example, a common source amplifier with resistive load as shown in Figure 3 was analyzed. The parameter values taken into consideration are R_d , W_0, L_0, W_1, L_1 and output values achieved as specification are gain (dB) and power (mW). The circuit was designed and simulated in 65nm CMOS technology in Cadence Virtuoso environment. A few sample points were generated and the dataset obtained was then interpolated as well as extrapolated to get more than 2500 sample points. The technique applied in this circuit is ANN with multiple activation functions used in series as a means to achieve negligible loss value.

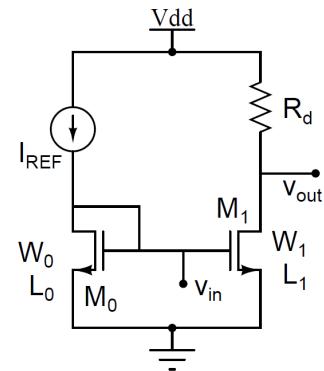


Fig. 3. Common Source Amplifier

In the above circuit, for a set of input parameters, value of I_d is taken as base value and gain is calculated in first row of Table I. The parameters are then varied so as to change the I_d value and hence changing the gain value. The comparison of manual calculation, SPICE output and machine learning model output is given in Table I by varying the values of I_d . It can be seen that the values of gain of common source amplifier circuit predicted, are very close to the manual calculation results as well as to the Cadence/SPICE output. This model operates on dataset containing more than 2500 values and gets trained in less than two minutes. This technique is based on both equation-based and simulation-based approaches, which make it difficult for application on

large circuits. Also, as mentioned above, giving parameters as inputs and taking specifications as outputs can be done with the help of SPICE softwares in nearly same time, hence this method may be used for the purpose of illustration.

TABLE I
OUTPUT COMPARISON FOR GAIN VALUES IN dB FROM VARIOUS SOURCES

I_d value	Gain Value		
	Manual Calculation	Cadence Output	Machine Learning Model
100%	4.10	4.05	4.09
200%	7.57	7.59	7.4
250%	10.0	10.2	10.07
300%	12	11.95	12.02

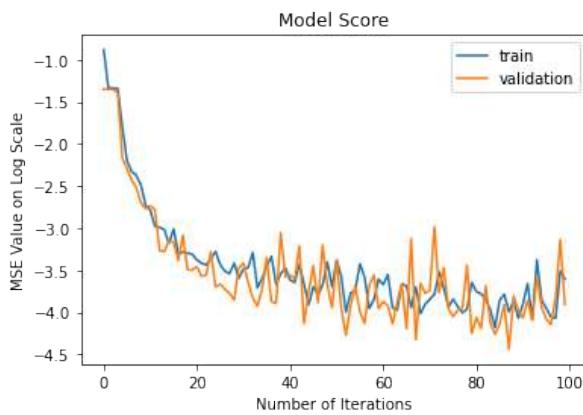


Fig. 4. Model MSE Plot as a function of Iterations

$$MSE = \frac{1}{n} \sum_{n=1}^n (|actual - predicted|^2) \quad (1)$$

Various combinations of activation functions, namely, Linear, ReLu and Softmax were used and best results in terms of MSE (can be calculated by Equation 1) obtained are shown in Figure 4. As shown in the figure, MSE value decreases by increasing the number of iterations for both training and validation dataset. The MSE value at 0th iteration is near -1dB and decreases to -4dB after 100 iterations with the stopping criterion of iterations as 0.1% of initial MSE value.

B. Backward Propagation Technique - Local ANN using GA based global optimization Technique

The supervised learning technique [3] combines the GA and ANN techniques to achieve an automated analog circuit design flow. Instead of training the ANN on a very large dataset, the paper proposes the training on a filtered dataset obtained as the output of GA. The algorithm is shown in Figure 5. A comparison of the GA-ANN model with other ML models namely, K-Nearest Neighbours (KNN), Decision Trees (DTs), Support Vector Machine (SVM), etc is performed. The algorithm in [3] starts with the global and local optimizations running alternatively until the performance metrics are reached or the maximum iteration limit is reached. The initial population of the GA-based algorithm is generated using the orthogonal array (OA) based Latin Hypercube Sampling (LHS). Parallel SPICE simulations are performed

to select the best candidate. The training dataset for the ANN model is generated from the local neighbourhood using OA-based LHS. Inside the ANN model, Local Minimum Search (LMS) is carried out and the optimum solution is verified using the SPICE simulations. A two-stage rail-to-rail Opamp and active-RC Chebyshev band-pass filter (CBPF) are used as examples. This method employing the Local-ANN using the parallel simulations generates an optimal circuit solution four times faster than the Global-ANN technique. A genetic algorithm based optimization is also needed in order to reduce the range of dataset over which the Local-ANN need to be trained.

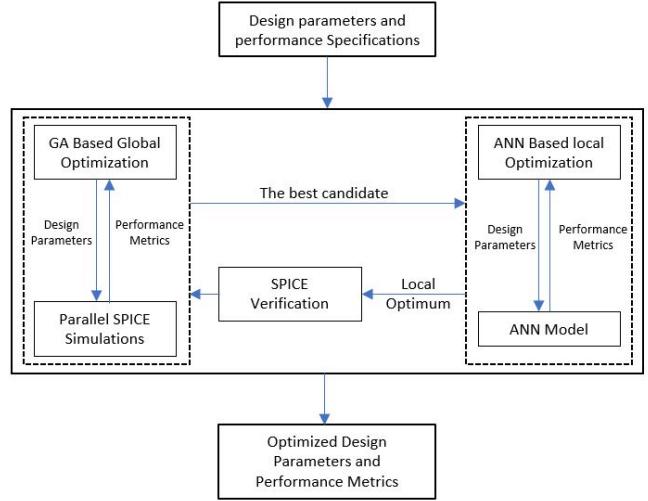


Fig. 5. Algorithm for Local ANN [3]

IV. PROPOSED MODEL USING GLOBAL ANN

In this section, we propose a new ML Model using global ANN. It consists of the input layer, an output layer, and hidden layers. The number of input neurons in the input layer depends on the input parameters of the ANN model. For example, for the design of an amplifier, the input parameters can be the performance metrics like DC gain, current (I_{dd}), gain-bandwidth product (GBW), etc. The number of output neurons depends upon the number of design parameters that we need to estimate, for example, the transistor widths, lengths, resistances, etc. Since there are multiple input and output parameters, we use a multi-layer perceptron (MLP) ANN model. The number of hidden layers and the number of neurons in each hidden layer in the architecture is a hyper-parameter that can be optimized through the validation process. The representative diagram for the ANN model is as shown in Figure 6.

Each layer of neurons is connected to other layer neurons through a fully connected network. The weights and bias represent the relation between each layer of data. These parameters are determined by minimizing the objective function which is the averaged error. The weights and biases are determined through the back propagation method. In the forward pass, the network will save all the intermediate derivatives. The optimum parameters for the ANN model are obtained by minimizing the loss function with respect to each weight and bias values using chain rule.

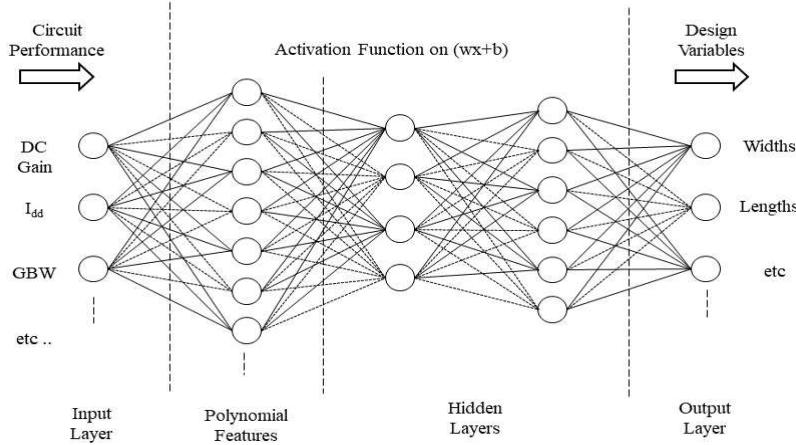


Fig. 6. ANN Network Model for Multi-Input Multi-Output System [4]

Figure 7 shows the flowchart of newly proposed ML implementation of the ANN algorithm. The input to the algorithm will be the desired specifications and output will be the set of design parameters. For the initial training of the ANN model the dataset collection is done using two different methods, W , L variation and I_d/W variation. In the first method, the width and the length of each transistor are varied between a range of maximum and minimum in nonlinear step size to get an independent dataset. This method gives a large dataset requiring a filtering step to remove all unwanted data which is similar procedure to the previous works in [3]. For example, the cases of non satisfying region of operation leading to a random behaviour of the response can be ignored. The second method proposed in this work generates the dataset based on the I_d/W values. These I_d/W values need to be varied in a small range only (as shown in Table II) as this range is chosen based on the simulations. This has an advantage of working with a smaller dataset which eliminates the filtering step. Once the dataset is obtained from the above two methods, 80% of data is used for training the ANN model and the remaining 20% is used for testing the accuracy of the trained model. To reduce the number of simulations for the initial analysis, the length of all the transistors is taken as the same. To train the ANN model, slew rate, gain, unity gain frequency, phase margin, and power is taken as the input parameters, and I_{bias} , V_{incm} and I_d/W or width and length values are taken as output parameters. The first step in training the ANN model is data visualization and pre-processing of the data values.

Some of the data values show a skewed pattern which can be simplified into a more evenly distributed dataset using the log and power transforms. If the ANN model is trained using these data values directly, it will not converge to a solution set as the data values are in different ranges. Hence, we need to normalize these data values before training of the ANN model. The operations performed on the training data has to be performed for the testing data as well. The MLP regressor model is used for the multi-input multi-output system. The MLP regressor is implemented using three hidden layers with ReLu activation function, adaptive learning rate, and Adam solver. The MLP regressor fits the model by minimizing the squared error objective function. The accuracy of the model is evaluated using the test data. The parameters slew

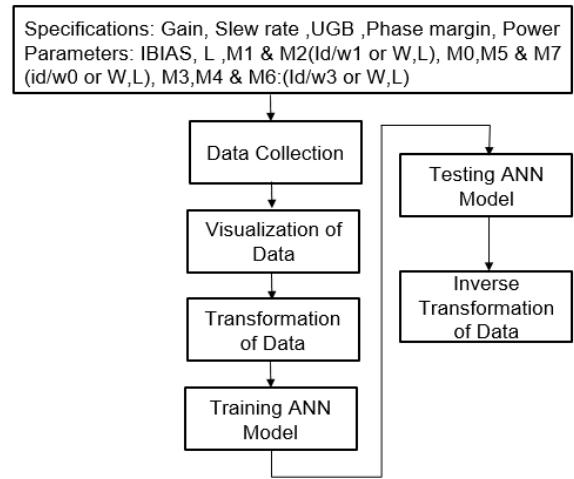


Fig. 7. Flowchart of ML Implementation of the ANN Algorithm

rate, gain, unity gain frequency, phase margin, and power from the test data are given as input to the trained ANN model, and the output values namely I_{bias} , V_{incm} and I_d/W or width and length are predicted. As the predicted data is in the normalized form, to bring back to the original range inverse transformations are applied to the predicted data. The predicted values are then compared with the actual values from the test data followed by the score and MSE calculation.

A two-stage single ended Opamp as shown in Figure 8, is implemented in 65nm CMOS process. The transistor widths, lengths, resistances, and capacitances are chosen as design parameters for optimization. The dataset for training the ANN model is generated using the g_m/I_d variation method and W , L variation method. Considering the matching criteria there will be a total of 13 independent parameters (W_0 , W_5 , W_7 , W_1 , W_3 , W_6 , L_0 , L_1 , L_3 , R_z , C_z , V_{incm} , I_{bias}). For example, the input transistors (M_1 , M_2) and (M_3 , M_4) are fully matched pairs so their W , L values are the same. The length of the transistors (M_0 , M_5 , M_7) and (M_3 , M_4 , and M_6) are taken to be L_0 and L_3 respectively. In the W , L varying method all the above said 13 parameters are varied to obtain a dataset of the performance metrics. The design parameters are varied from a minimum value to a maximum value with

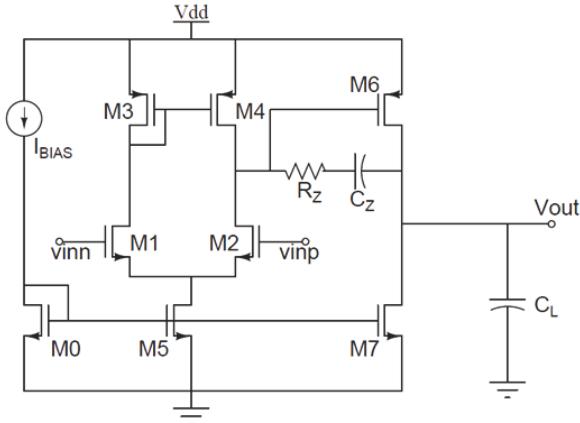


Fig. 8. Two-Stage Single-Ended Opamp

a nonlinear step size. In the case of the g_m/I_d method the transistor sizes are determined from the current density (I_d/W) plots obtained from the Cadence simulations for different lengths. The overdrives of the transistors are chosen such that they operate in a strong inversion region, then the corresponding I_d/W values are taken. For the generation of the dataset to train the ANN model, the parameter varied in the g_m/I_d method is the current density (I_d/W). Since the overdrives of the transistors M_0 , M_5 , M_7 are the same their I_d/W values are changing in a similar manner. M_1 , M_2 and M_3 , M_4 , M_6 combinations follow the same pattern. I_{bias} is kept as a variable to change the current through these transistors. The operational amplifier tail current is decided as a scaled factor of I_{bias} . The current through the input transistors is taken as half of the tail current and the output branch (M_6 , M_7) is taken as 10 times the current of input transistors. Thus the total number of independent variables in this case are 11 (I_d/W_0 , I_d/W_n , I_d/W_p , I_{bias} , I_{tail} , V_{incm} , R_z , C_z , L_0 , L_1 , L_3). The range of I_d/W values is given in Table II. The other parameters are varied from a minimum to a maximum with a nonlinear step size.

TABLE II
 I_d/W RANGE OF NMOS AND PMOS TRANSISTORS FOR DIFFERENT LENGTHS

$L(\mu m)$	I_d/W_n	I_d/W_p	I_d/W_0
0.165	[24.4, 51.6]	[3.685, 10.74]	[20.3, 88.58]
0.565	[4.661, 13.35]	[1.061, 3.158]	[4.661, 24.72]
1.056	[2.56, 7.34]	[0.61, 1.77]	[2.56, 13.67]
5.056	[0.5, 1.512]	[0.113, 0.34]	[0.5, 3.27]

In the unsupervised learning technique mentioned in [2], the simulations are performed for a three-stage trans-impedance amplifier and a two-stage trans-impedance amplifier. The dataset size used in training the ML model is given in Table III. The data samples required is larger than human expert model. Comparing with the human expert model the three-stage trans-impedance amplifier and two-stage trans-impedance amplifier gives a sample efficiency of 250 and 25 respectively. In the supervised learning technique [3], the simulations are performed for two-stage rail-to-rail Opamp and active-RC CBPF. It uses filtered data from the GA algorithm to generate training data for the local ANN model. In the proposed model of global ANN, which uses an

optimised data generated using the g_m/I_d technique performs the simulation on a two stage single ended Opamp. The score was obtained as 93.44% and an MSE of 0.055. The score values for W, L method and proposed g_m/I_d method show that the trained ANN model accurately predicts the data with greater than 90% accuracy. Comparison between the proposed global ANN using g_m/I_d method and W,L varying method shows that the time required to generate the dataset using cadence simulations is substantially less. Though the dataset size is very small, the proposed method gives accurate results as the parameters which are varied to generate the dataset are simulation based.

TABLE III
COMPARISON TABLE OF DIFFERENT ML TECHNIQUES

Method	Dataset	Score(%)
Unsupervised [2]	40000 - 50000	-
Global ANN - Supervised [3]	45583	30.2
Local ANN - Supervised [3]	632	75.8
This work - W, L - Supervised	65534	90
This work - g_m/I_d - Supervised	7361	93

V. CONCLUSION

This paper analyses two methods of automated analog circuit design using machine learning techniques, namely the forward propagation method and backward propagation method. In forward propagation we are able to predict the expected performance metrics for the given design parameters, while in the backward propagation method we are predicting the design values for a given specifications. The backward propagation method uses a novel data acquiring technique using the g_m/I_d method which reduces the dataset size and time for analysis significantly. The new technique proposed in this paper shows that the global ANN model operates on smaller dataset size efficiently. It trains the ANN model with fewer hidden layers and the number of neurons to produce optimum design parameters. In the (W, L) variation method, the simulation time to generate the training data for the ANN model is quite long. In g_m/I_d supervised method, the dataset size is effectively limited as the simulations are done only for the selected region of operation. The time taken, dataset size and the efficiency is higher for the g_m/I_d method as it considers the simulation-based sample space for dataset generation. This method can be extended further for design of complex analog circuits.

REFERENCES

- [1] G. Wolfe and R. Vemuri, "Extraction and use of neural network models in automated synthesis of operational amplifiers," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 22, no. 2, pp. 198–212, Feb. 2003
- [2] H.Wang, J.Yang, H.S.Lee, S.Han, "Learning to design circuits," 32nd Conference on Neural Information Processing Systems , Montréal, Canada, 2018.
- [3] Y. Li, Y. Wang, Y. Li, R. Zhou and Z. Lin, "An Artificial Neural Network Assisted Optimization System for Analog Design Space Exploration," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 10, pp. 2640-2653, Oct. 2020, doi: 10.1109/TCAD.2019.2961322.
- [4] J.Rosa, D.Guerra, N.Horta, M. F. Martins, N.Lourenco, "Using Artificial Neural Networks for Analog Integrated Circuit Design Automation", Springer Briefs in Applied Sciences and Technology, <https://doi.org/10.1007/978-3-030-35743-6.hed>.

- [5] M. Kotti, M. Fakhfakh and E. Tlelo-Cuautle, "Effect of the design space sampling on the design performances," 2018 IEEE 9th Latin American Symposium on Circuits & Systems (LASCAS), 2018, pp. 1-4, doi: 10.1109/LASCAS.2018.8399926.
- [6] Ethem Alpaydin, "Introduction to Machine Learning" Second Edition, The MIT Press, 2010.
- [7] <https://towardsdatascience.com/deep-deterministic-policy-gradients-explained-2d94655a9b7b>
- [8] M. I. Dieste-Velasco, M. Diez-Mediavilla and C. Alonso-Tristán, "Regression and ANN Models for Electronic Circuit Design", Hindawi Complexity Volume 2018, Article ID 7379512, doi.org/10.1155/2018/7379512.
- [9] Engin Afacan, Nuno Lourenço, Ricardo Martins, GÜnhan Dündar, "Review: Machine learning techniques in analog/RF integrated circuit design, synthesis, layout, and test", INTEGRATION, the VLSI journal 77 (2021) 113–130 ELSEVIER