
Macro-modeling of OTA using ANN for fast synthesis

B. Shivalal Patro*
Sushanta K. Mandal**

Abstract

This paper proposes performance based macro modeling of operational transconductance amplifier (OTA) using artificial neural network (ANN). ANN models the analog circuits and provides a relation between input and output parameters. In this work, OTA output parameters such as gain and common mode rejection ratio (CMRR) are modeled as a test circuit which is designed in Cadence Virtuoso on UMC 180 nm technology. The width of the transistors are taken as the input parameters of the ANN. From the SPICE simulation results, the feasible dataset has been extracted from the complete dataset. The final results show that accurate neural models are generated successfully.

Keywords:

Analog Circuit;
Machine Learning;
Macro modeling;
Mean Square Error;
Regression Analysis.

Author correspondence:

Sushanta K. Mandal,
Department of Electronics and Communication Engineering
CUTM University, Bhubaneswar, Odisha, India

1. Introduction

The analog and digital circuits are the main constituents of the integrated circuits (ICs). The demand of very high performance, smart devices having all the features to perform multi-tasks is due to improve the quality of life and speed. Day by day, the complexities of the ICs are increasing due to the increase in number of transistors in a system or a chip according to Moore's law [1], [2]. The exponentially increasing transistors and with very less time to market, designing an optimized circuit for a specific task has become tedious jobs for the designers. There are many design and verification tools available but they are not able to give the feasible solution in desired limited time [3]. One of such powerful tool is SPICE based simulation modeling. It is a very powerful simulation tool which performs many type of analysis with very high accuracy [4]. But with growing complexities of the circuits and considering the ever-decreasing time to market, this type of simulation methods is not able to cope up. In earlier days, where there are no advanced circuit simulators, at that time circuit designers depend upon the equation based analysis of the circuits. Hence, the designers need to be expertise in the field of circuit design and analysis. So, the equations formulated are highly accurate. But with the growing complexities of the circuits, the equation formulation becomes more complicated and time consuming. This led to more design error. So, some automated techniques have been applied in order to reduce the errors and time. But still it cannot match the quick time to market.

So, the circuit designers started developing macro-models for the circuits using various techniques which will be used for quick synthesis of circuits. The most important and such type of reliable method used for modeling of the analog circuit is by simulation based approaches. The designers rely upon the general circuit simulator like SPICE to acquire the performance of the circuits. These simulators generate the feasible models which are used for synthesis. Similar work is discussed in [5]. But this modeling method requires very large time especially for the complicated circuits. This will lead to more design time for synthesis of the circuits. Another macro-modeling method which is widely used by many VLSI designers is symbolic technique [6]–

*B. Shivalal Patro, School of Electronics Engineering, KIIT University, Bhubaneswar, Odisha, India, Email: shivalalpatro@gmail.com

**Sushanta K. Mandal, Department of Electronics and Communication Engineering, CUTM University, Bhubaneswar, Odisha, India, Email: sushantakumar@yahoo.com

[9]. In these papers, the authors try to estimate the performance parameters by forming the circuit model equations. For automated symbolic techniques [10]–[13]. But the final equations generated are complicated. This equation can be simplified by compromising the accuracy by reducing the less affecting terms. Using the help of global optimization techniques like geometric programming [14]–[16], posynomial based behavioral models are generated. Similar to symbolic methods, this technique also goes through generation of equations. The main advantage of geometric programming is that it is very fast and more accurate than symbolic methods. But it needs more in-depth knowledge of the circuits. Also, sometimes the manual formulation of the posynomial constraints especially during the synthesis of the circuits are mandatory. There are some papers [17] which try to automate the formulation of the equations but affects the accuracy of the models to a larger extent. Model Order reduction also works in the similar aspect for generating the mathematical equations for the analog circuits [18]–[21]. It takes the help of SPICE generated netlist data and formulate the equations and automatically. It has a very good merit of handling complex circuits. It approximates the circuits and reduces the large number of equations into reduced numbers in the form of reduced models. Still the design time formulation will require a lot more time while implementing the models for synthesis purpose which requires a lot of iteration.

In last decades, neural networks have been widely used for modeling for synthesis of analog circuit designs especially circuits operating in the RF and microwave range. These are black box models and don't describe the qualitative aspect of the internal circuits. But they can formulate nonlinear and hyper-dimensional functionalities at a very fast rate. In [22]–[24] generation of the models from the S-parameter has been discussed. In [25]–[27], similar models are generated to identify the problems and synthesis respectively.

2. ANN based Macro-modeling

The growing demand for automation has led the researchers in the area of the circuit design to implement similar methods for formulating the models for the analog circuits. Artificial neural network (ANN) is one of such advanced technique which can be used to generate the macro-models. Neural networks are referred as the connection systems inspired by the biological neural networks, which are found in human and animal brains and bodies. Similar to the human brain, ANN learns progressively by providing some learning algorithms. Typically, ANN constitutes artificial neurons are arranged in layers. These neurons act as storing the information in the form of numbers, especially 0 and 1. Also, they transfer the information to neurons of other layers with suitable weights.

As shown in Figure 1, ANN consists of three main layers:

- Input Layer: Layer 1
- Hidden Layer: Layer 2 ... Layer L-1
- Output Layer: Layer L

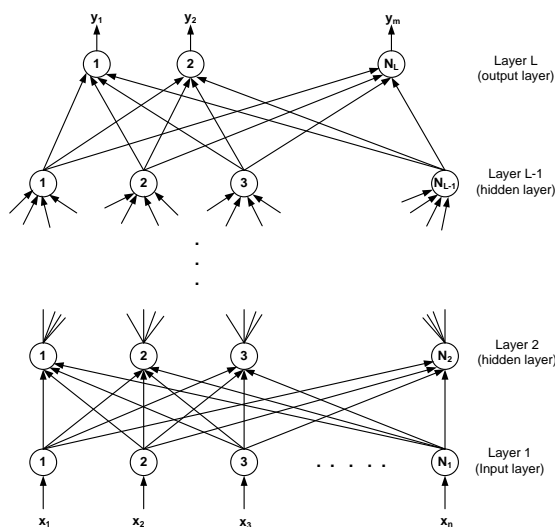


Figure 1: *Artificial Neural Network (ANN) structure*

Input layer composed of input variables. Hidden layer composed of hidden neurons with calculated weights. They can be multi-layers. Output Layer composed of output parameters of the model. They can be one for a single output and many numbers of nodes for multi-output model based on macro-modeling technique.

A basic ANN composed of three types of parameters:

- Connection pattern between the layers of the neural network
- Weights of the connections
- Activation function

The connection pattern is referred by the parameter 'bias'. The weights of the connections are updated during the training of the neural network, which is also called as a learning process. The activation function converts the neuron's weighted input to its output activation. There are many applications of artificial neural network in the field of classifications and curve fitting problems [28], [29]. Similarly, it has been widely used in the field of analog and mixed signals. The main objective of the formation of the neural network model is to provide a black box model which can be useful for reducing the synthesis time for the circuits. A neural network model is generated from the discrete sets of data. These data sets are called as training data sets. These data are representations of the functionalities of the system which is to be modeled. So, it is not mandatory to be aware of the system in detail. ANNs have the capability to learn from any arbitrary set of non-linear input-output dataset. Mathematically, a neural network can be represented by a function let $f(x)$ and can be defined as

$$f(x) = K\left(\sum_i w_i g_i(x)\right) \quad (1)$$

Here, 'K' means the activation function. 'w' represents the bias weight of each network 'i', 'g(x)' represents the other functions which can be considered as input functions obtained from the previous layer. Here the weights are generated in a probabilistic manner. The most important step of generating the model is the training procedure. There are many training algorithms available like backpropagation, conjugate-gradient, quasi-Newton, Levenberg-Marquardt, etc. Table I provides the ANN performance of the different macro-models generated using different training algorithms. From the results it is observed that LM algorithm provides better results. So, in this paper, Levenberg-Marquardt (LM) back propagation algorithm is taken for the training of ANN models. This network training function updates weight and bias states according to Levenberg-Marquardt optimization technique [30], [31].

Table 1. ANN Parameters during modeling of test circuits

Sl. No.	Training Algorithm	Gain Model		CMRR Model	
		Total MSE	Total Regression	Total MSE	Total Regression
1	BFGS Quasi-Newton	0.00346	0.99182	0.00196	0.99533
2	Conjugate Gradient with Beale Powell Restarts	0.00302	0.99287	0.00474	0.98868
3	Gradient Descent with Momentum	0.10094	0.73862	0.10365	0.72206
4	Levenberg Marquardt	0.00029	0.99932	0.00026	0.99936
5	RProp	0.01359	0.96798	0.00344	0.99189

Table 2. ANN Parameters during modeling of test circuits

Sl. No.	ANN Parameters	Remarks
1	No. of Hidden Layers	1
2	Total no. of layers from input to output	2
3	Activation function for Hidden Neurons	Sigmoid function
4	Activation function for Output	Linear function
5	Total no. of hidden neurons	20
6	Training Algorithm	Levenberg-Marquardt
7	Number of inputs for validation	10%

This training algorithm is preferred as it is one of the fastest training methods in ANN model generation. It has got a demerit that it uses a huge memory space and hence the model generation becomes very slow when complex or huge data sets are taken for modeling. For generating ANN model for the two test circuits, the parameter settings of ANN are shown in Table 2. The model is generated using the help of MATLAB platform using Neural Network toolbox (NF tool) [32]. A two-layer feed-forward network with sigmoid hidden neurons and linear output neurons (fitnet), can fit multi-dimensional mapping problems,

arbitrarily well, given consistent data and enough neurons in its hidden layer. Validation of the models is used to measure network generalization, and to halt training when generalization stops improving.

3. Simulation Results

3.1. OTA Simulation Setup

A single output two-stage operational transconductance amplifier (OTA) is shown in Fig. 2. The circuit has eight MOS transistors, compensation capacitance and a reference bias current. This circuit is designed in UMC 180nm based technology in Cadence Virtuoso environment. The transistors widths are varied from 350 nm to 1500 nm design space of 5 steps equally distributed. The performance parameters gain and CMRR contain a wide range of values. If this wide range of values is taken for modeling purposes, then the model accuracy will not be good. So, the non-required data sets are removed from the extracted datasets. The widths of the transistors PM1 and PM6 are matched to 'w1' and PM8 and PM9 are matched to 'w2'. Similarly, the widths of NM3 and NM4 are matched to 'w3'. This is done in order to evenly split the bias current provided to the differential pair by transistor PM6. Transistor PM7 width is denoted by the variable 'w4' and for NM2 is by 'w5'. Based on this five number of variables are used modeling of the output parameters (i.e. Gain and CMRR) of the transistors.

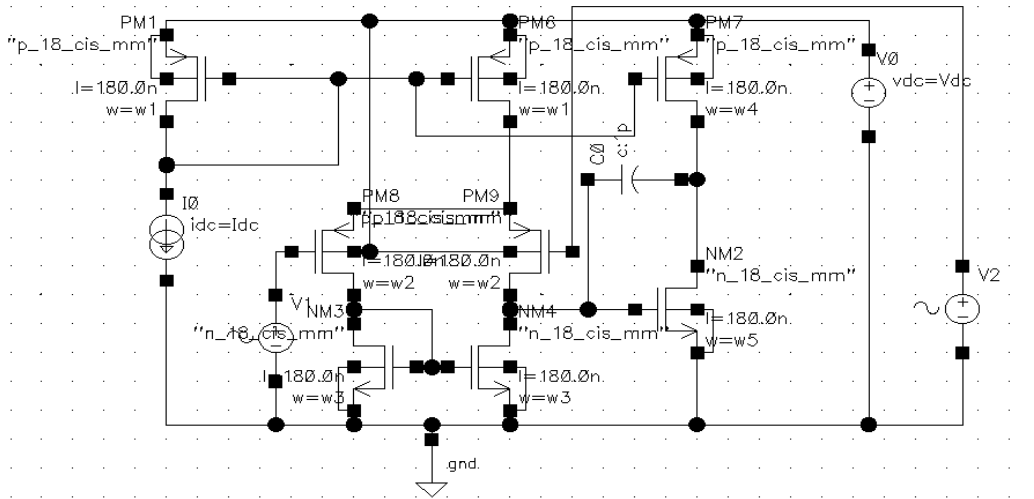


Figure 2. A two-stage Operational Transconductance Amplifier (OTA)

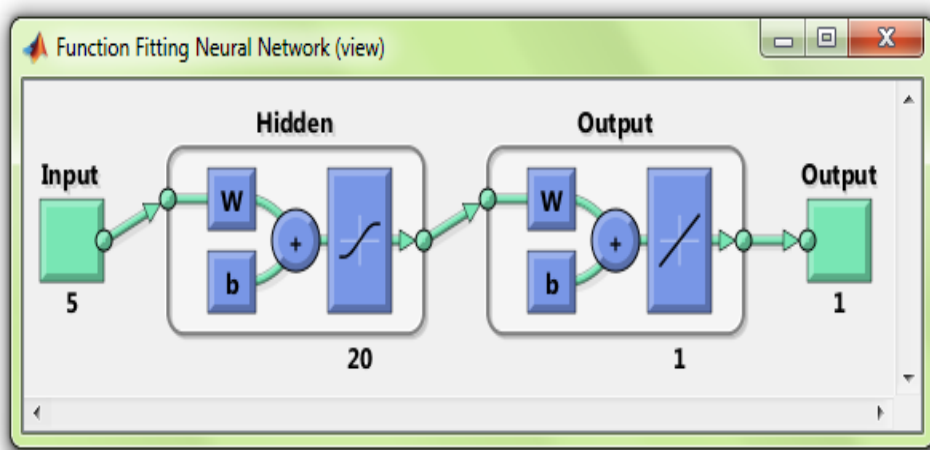


Figure 3. Neural Network Model generated for modeling of OTA

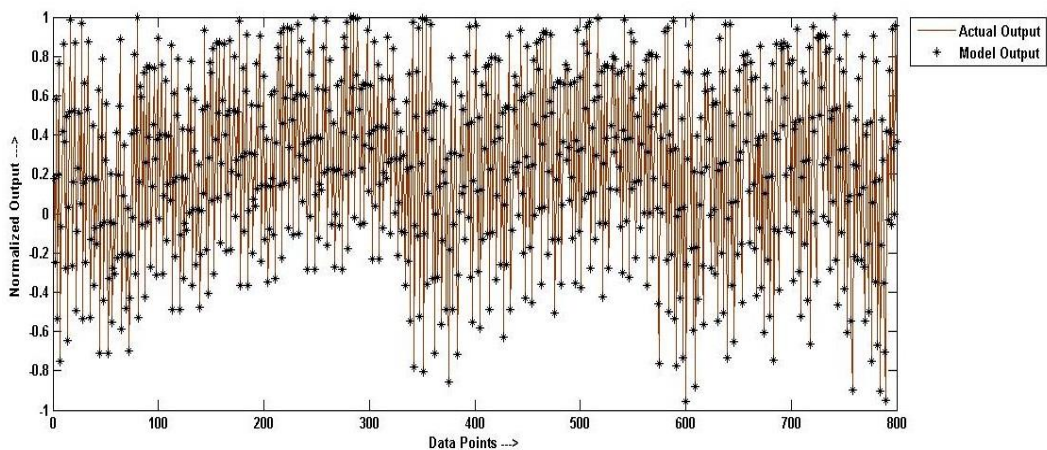


Figure 4. Correlation graph of actual Gain of the OTA and estimated Gain by the ANN model

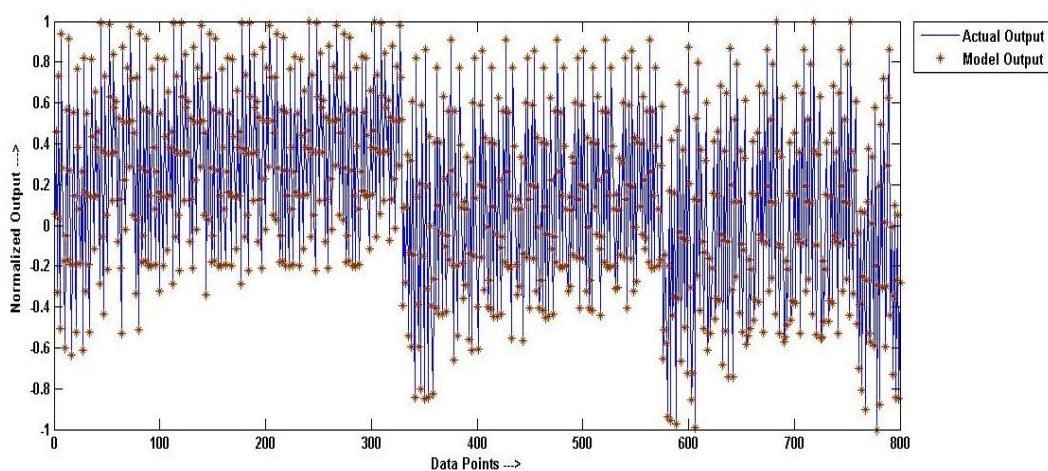


Figure 5. Correlation graph of actual CMRR of the OTA and estimated CMRR by the ANN model

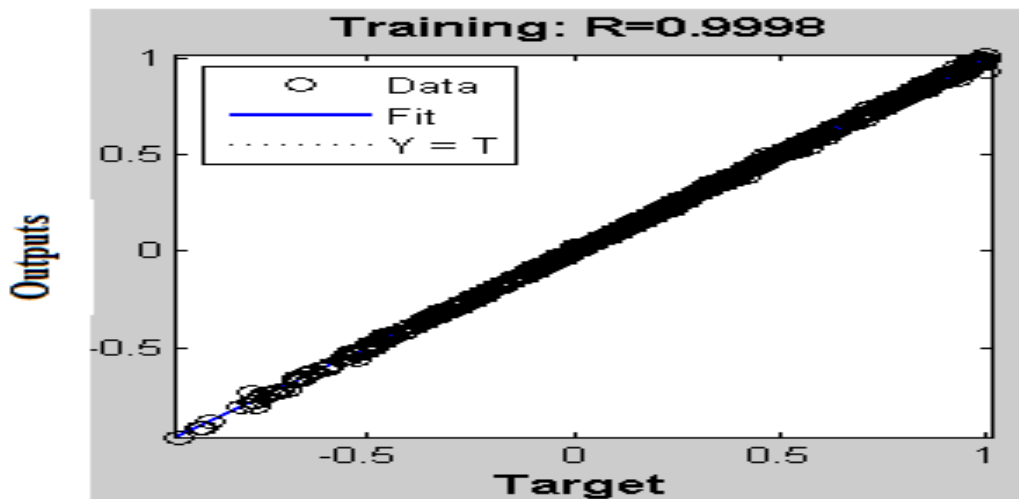


Figure 6. Scatter Plot for normalized model outputs and the actual output of Gain of OTA during training period

3.2. Simulation and Results Analysis

Figure 3 shows the neural network fitting model generated for modeling the output parameters of OTA. It can be observed that there are 5 numbers of input nodes, which are the widths of the transistors. There is a single hidden layer with 20 number of hidden neurons and then one output node for one output. Figure 4 and Figure 5 show the correlation output of the actual gain and actual CMRR with respect to the ANN model respectively. The correlation graphs show a very good mapping of actual output and the targets generated after modeling by the ANN model. Figure 6 and Figure 7 show the training and testing regression values of gain of OTA. Figure 8 and Figure 9 show the training and testing regression values of CMRR of OTA. The values are nearly 1 which show a good mapping of actual targets and model outputs.

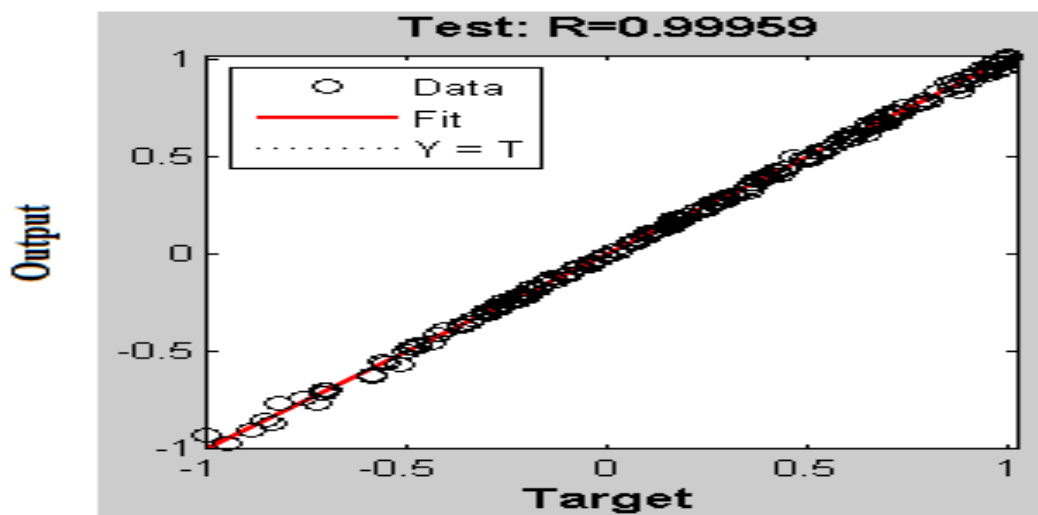


Figure 7. Scatter Plot for normalized model outputs and the actual output of Gain of OTA during testing period

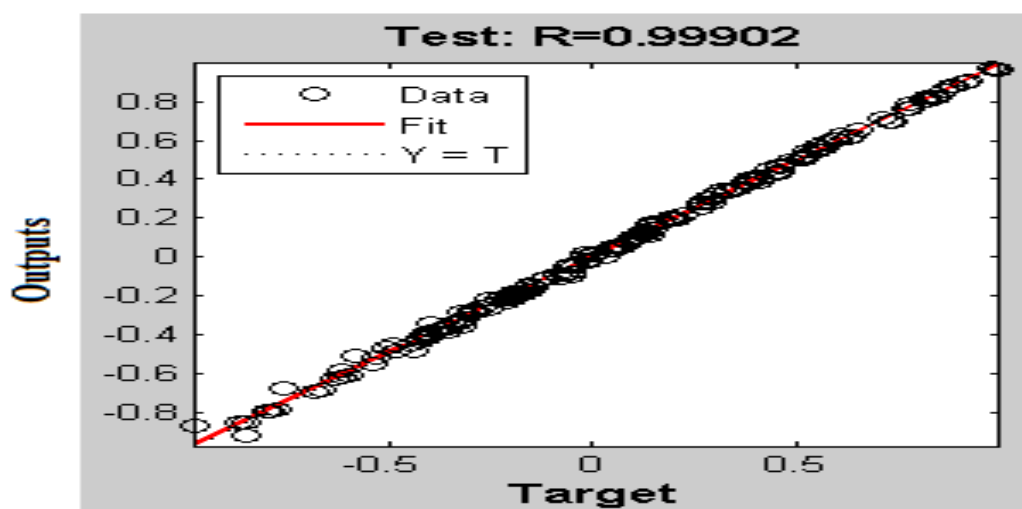


Figure 8. Scatter Plot for normalized model outputs and the actual output of CMRR of OTA during training period

The detailed results of ANN modeling of OTA are shown in Tables III and IV. From the gain model, during the training period, the mean MSE obtained is 0.024% and regression value is 99.94%. Similarly, during testing period the mean MSE obtained is 0.039% and regression value is 99.9%. Similarly, for CMRR model of OTA, the mean MSE was found to be 0.043% and regression value was obtained as 99.9% during the training period. Again during the testing period, the mean MSE obtained is 0.092% and the regression value was obtained as 99.8%. This shows that the OTA output parameters, i.e. Gain and CMRR have been modeled with a very good accurate value.

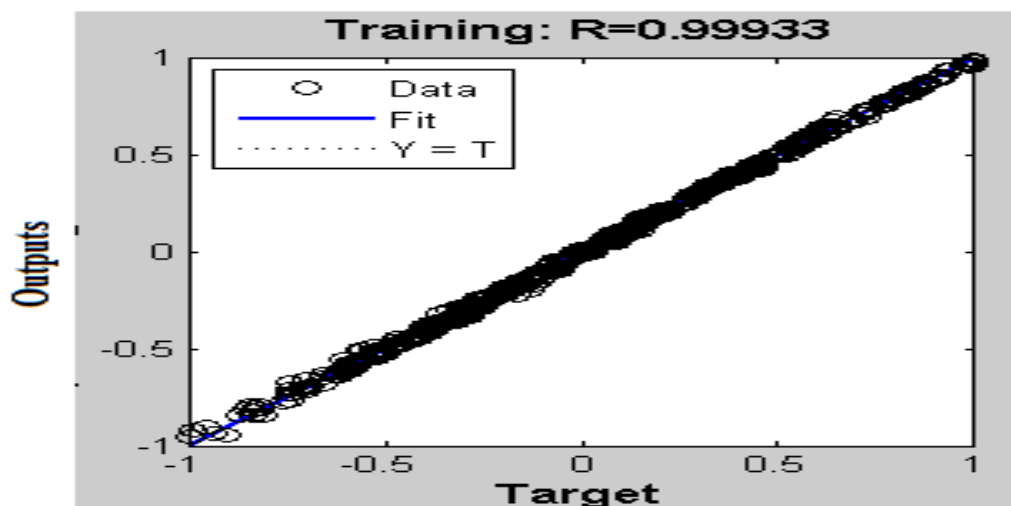


Figure 9. Scatter Plot for normalized model outputs and the actual output of CMRR of OTA during testing period

Table 3. ANN Modeling of Gain for OTA

No. of runs	No. of Epochs	Training Data		Testing Data	
		MSE	Regression	MSE	Regression
1	115	0.00021	0.99946	0.00025	0.99923
2	92	0.00037	0.99902	0.00052	0.99855
3	69	0.00037	0.99901	0.00068	0.99822
4	74	0.00011	0.99972	0.00021	0.99935
5	125	0.00017	0.9998	0.00031	0.99959
Mean	95	0.00024	0.999402	0.00039	0.998988

Table 4. ANN Modeling of CMRR for OTA

No. of runs	No. of Epochs	Training Data		Testing Data	
		MSE	Regression	MSE	Regression
1	70	0.00048	0.99889	0.00101	0.99771
2	77	0.00018	0.99958	0.00034	0.99918
3	27	0.00081	0.99811	0.00137	0.99668
4	111	0.00042	0.99899	0.00099	0.99802
5	81	0.00025	0.99933	0.00087	0.99902
Mean	73.2	0.000428	0.99898	0.000916	0.998122

4. Conclusion

This paper discusses macro-modeling of OTA performance parameters i.e. gain and CMRR using ANN. The parameters have been accurately modeled which can be used in synthesis of the OTA based on desired specification at a very quick rate as compared to the circuit simulator. This will reduce the design turn around time. Using this technique neural models of other circuit parameters can be developed for fast synthesis of the circuits.

References(12pt)

- [1] R. R. Schaller, "Moore's law: past, present and future," *IEEE Spectr.*, vol. 34, no. 6, pp. 52–59, 1997.
- [2] B. S. Patro and B. Vandana, "Low Power Strategies for beyond Moore's Law Era: Low Power Device Technologies," *Des. Model. Low Power VLSI Syst.*, p. 27, 2016.
- [3] W.-K. Chen, *Analog and VLSI circuits*. CRC Press, 2009.
- [4] F. N. Najm, *Circuit simulation*. John Wiley & Sons, 2010.
- [5] H. A. Hamad and R. E. Massara, "A macromodeling approach to the synthesis of analog circuits using circuit decomposition and numerical circuit analysis," in *Circuits and Systems, 1992., Proceedings of the 35th Midwest Symposium on*, 1992, pp. 1108–1111.

- [6] G. G. E. Gielen, H. C. C. Walscharts, and W. M. C. Sansen, "Analog circuit design optimization based on symbolic simulation and simulated annealing," *IEEE J. Solid-State Circuits*, vol. 25, no. 3, pp. 707–713, 1990.
- [7] K. Swings, G. Gielen, and W. Sansen, "An intelligent analog IC design system based on manipulation of design equations," in *Custom Integrated Circuits Conference, 1990., Proceedings of the IEEE 1990*, 1990, pp. 6–8.
- [8] G. Gielen, P. Wambacq, and W. M. Sansen, "Symbolic analysis methods and applications for analog circuits: A tutorial overview," *Proc. IEEE*, vol. 82, no. 2, pp. 287–304, 1994.
- [9] C.-J. Shi and X.-D. Tan, "Canonical symbolic analysis of large analog circuits with determinant decision diagrams," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 19, no. 1, pp. 1–18, 2000.
- [10] T. McConaghy and G. G. E. Gielen, "Template-free symbolic performance modeling of analog circuits via canonical-form functions and genetic programming," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 28, no. 8, pp. 1162–1175, 2009.
- [11] E. Tlelo-Cuautle, C. Sánchez-López, E. Martínez-Romero, and S. X.-D. Tan, "Symbolic analysis of analog circuits containing voltage mirrors and current mirrors," *Analog Integr. Circuits Signal Process.*, vol. 65, no. 1, pp. 89–95, 2010.
- [12] C. Ferent and A. Doboli, "Symbolic matching and constraint generation for systematic comparison of analog circuits," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 32, no. 4, pp. 616–629, 2013.
- [13] M. Shokouhifar and A. Jalali, "An evolutionary-based methodology for symbolic simplification of analog circuits using genetic algorithm and simulated annealing," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1189–1201, 2015.
- [14] M. del Mar Hershenson, S. P. Boyd, and T. H. Lee, "GPCAD: A tool for CMOS op-amp synthesis," in *Proceedings of the 1998 IEEE/ACM international conference on Computer-aided design*, 1998, pp. 296–303.
- [15] P. Mandal and V. Visvanathan, "CMOS op-amp sizing using a geometric programming formulation," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 20, no. 1, pp. 22–38, 2001.
- [16] J. Kim, L. Vandenberghe, and C.-K. K. Yang, "Convex piecewise-linear modeling method for circuit optimization via geometric programming," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 29, no. 11, pp. 1823–1827, 2010.
- [17] L. Hannah and D. Dunson, "Ensemble methods for convex regression with applications to geometric programming based circuit design," *arXiv Prepr. arXiv1206.4645*, 2012.
- [18] P. Li and L. T. Pileggi, "Compact reduced-order modeling of weakly nonlinear analog and RF circuits," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 24, no. 2, pp. 184–203, 2005.
- [19] H. Aridhi, M. H. Zaki, and S. Tahar, "Towards improving simulation of analog circuits using model order reduction," in *Proceedings of the Conference on Design, Automation and Test in Europe*, 2012, pp. 1337–1342.
- [20] R. A. Rutenbar, G. G. E. Gielen, and J. Roychowdhury, "Hierarchical modeling, optimization, and synthesis for system-level analog and RF designs," *Proc. IEEE*, vol. 95, no. 3, pp. 640–669, 2007.
- [21] J. Roychowdhury, "Algorithmic macromodelling methods for mixed-signal systems," in *VLSI Design, 2004. Proceedings. 17th International Conference on*, 2004, pp. 141–147.
- [22] G. L. Creech, B. J. Paul, C. D. Lesniak, T. J. Jenkins, and M. C. Calcaterra, "Artificial neural networks for fast and accurate EM-CAD of microwave circuits," *IEEE Trans. Microw. Theory Tech.*, vol. 45, no. 5, pp. 794–802, 1997.
- [23] Y. Fang, M. C. E. Yagoub, F. Wang, and Q.-J. Zhang, "A new macromodeling approach for nonlinear microwave circuits based on recurrent neural networks," *IEEE Trans. Microw. Theory Tech.*, vol. 48, no. 12, pp. 2335–2344, 2000.
- [24] F. Gong, H. Yu, and L. He, "Stochastic analog circuit behavior modeling by point estimation method," in *Proceedings of the 2011 international symposium on Physical design*, 2011, pp. 175–182.
- [25] C. Yuan, "Conditional multi-output regression," in *Neural Networks (IJCNN), The 2011 International Joint Conference on*, 2011, pp. 189–196.
- [26] L. Yuan, Y. He, J. Huang, and Y. Sun, "A new neural-network-based fault diagnosis approach for analog circuits by using kurtosis and entropy as a preprocessor," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 3, pp. 586–595, 2010.
- [27] G. Wolfe and R. Vemuri, "Extraction and use of neural network models in automated synthesis of operational amplifiers," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 22, no. 2, pp. 198–212, 2003.
- [28] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.
- [29] N. Gupta, "Artificial neural network," *Netw. Complex Syst.*, vol. 3, no. 1, pp. 24–28, 2013.
- [30] J. J. Moré, "The Levenberg-Marquardt algorithm: implementation and theory," in *Numerical analysis*, Springer, 1978, pp. 105–116.
- [31] M. I. A. Lourakis, "A brief description of the Levenberg-Marquardt algorithm implemented by levmar," *Found. Res. Technol.*, vol. 4, no. 1, 2005.
- [32] H. Demuth, M. Beale, and M. Hagan, "Neural network toolbox 6," *User's Guid.*, pp. 37–55, 2008.