Accelerating Voltage-Controlled Oscillator Sizing Optimizations with a Convergence Classifier & Frequency Guess Predictor

João Domingues
Instituto de Telecomunicações
Instituto Superior Técnico – Universidade de Lisboa, Lisboa, Portugal
joaoluisdomingues@tecnico.ulisboa.pt

***Abstract*— The work presented in this report belongs to the scientific field of electronic design automation, with a special focus on the automatic sizing of radio-frequency integrated circuit blocks. With the help of deep learning and, more specifically, artificial neural networks, a new approach is introduced and discussed. The approach proposed in this work is based on a supervised learning scenario using artificial neural networks both for classification and regression. A convergence classifier will be used to predict if a certain simulation is likely to converge or not, and a frequency guess predictor to predict the oscillating frequency. This method will be implemented and tested on voltage-controlled oscillators' optimizations, by learning from a dataset of previous performances obtained by the simulator. The studied voltage-controlled oscillator circuit topologies are evaluated under extreme operation, i.e., Process, Voltage and Temperature corners. It is expected that these networks filter and discard solutions with no valuable information for the optimization loop, and thus, greatly reducing the overall time of the optimization process. The result is a model that can predict non desired solutions, resulting in gains of almost 20% in overall simulation time, by discarding non valuable points. And it is able to correctly predict oscillatory frequencies, as the difference between these ones and the ones given by the simulator, reach values of mean absolute percentage error under 12%. Additionally, the use of this model does not compromise the results, as the ones obtained are very similar to the ones obtained without its use, and even better in some cases. The model demonstrates to be feasible for different optimization specifications, as well as for other examples of Voltage-Controlled Oscillators. The gains are similar, as the model is able to save 10% and 17%, respectively, and the results very promising, resulting in a model with a strong level of generalization**

Keywords—Artificial Neural Networks, Analog Integrated Circuits, Electronic Design Automation, Physical Design

I. INTRODUCTION

Electronic Design Automation (EDA) tools and design methodologies have been made available to cope with new capabilities offered by the integration technologies. However, there is still a huge discrepancy between the tools available for analog and digital integrated circuit (IC) design. The gap between the number of existing EDA tools for digital and analog circuits is usually explained by the fact that the digital market is much larger, absolving the available resources. It is also easier to express a digital system, which can be represented naturally in terms of Boolean representation, whereas, on the analog side, their design is less systematic, more knowledge-based, and more heuristic [1]. Even though analog circuits only occupy a small fraction of systems, they are responsible for design errors and expensive reruns. Therefore, economic pressure has motivated the pursuit of better methodologies to accelerate analog design.

The automation level for analog IC has been improving in the last years, being a field of profound academic and industrial research activity, which produces significant advances [2]. However, it is still far from the push-button stage, which leads to designers exploring the solution space almost manually as there are no standard advanced analog EDA tools and methodologies to automate the analog IC design flow.

On top of that, with predictions that more than half of new businesses will run on the Internet of Things (IoT) and advances in telecommunications, such as the 5th generation broadband or 5G for short, there will be a huge demand for devices and sensors, opening doors to advances in areas such as healthcare, education, resource management, transportation, agriculture, and many other areas. Not only that, but there has also been an increase in the amount of data that is being continuously generated, resulting in new challenges within every part of the networks. Consequently, there is high pressure in today's market for large communication rates, extensive bandwidths, and ultralow-power consumptions. This is where radio-frequency (RF) ICs come in hand, playing a crucial role. This demand stresses out the problem which resides in the remarkable difficulty of RF and Millimeter Wave (mmWave) IC design in deep nanometric integration technologies for both IoT and 5G, due to their high complexity and demanding performances. Aggravated by the need to fulfill these at minimal costs and under frightful time-to-market constraints. Some of the design difficulties lie in the exceptionally wide range of frequencies and dynamic ranges involved, but also:

- Their dependence on non-reliable models of passive devices;
- At gigahertz frequencies, there is a huge impact of layout parasitics;
- Their integration in deep nanometer technologies that are bearing variability issues and non-idealities which have never been experienced in older technology nodes.

One major objective for the design of modern RF ICs is to avoid the costs of redesign cycles, to diminish the post-

fabrication tuning, and pursue first-pass fabrication success. Until now, the circuit designers were able to carry this flow manually thanks to the vast Computer Aided Design (CAD) tools provided by companies, but this method is no longer manageable due to the number of complex interactions and the sub-optimal RF designs that come with it.

Over the past few years, the entire IC implementation from digital flow to graphic design system design flows, where one uses primarily stand-alone synthesis, placement, and routing algorithms to IC construction and analysis flows for design closure, has gone through significant changes. When it comes to analog design flow, most of the time when designing an Analog Mixed Signal (AMS) IC manually, the designer still follows the steps introduced by Gielen and Rutenbar [3], illustrated in Figure 1. The methodology consists of a series of top-down design steps that are repeated from the system-level to the device-level, and bottom-up layout generation and verification.

This document is organized as follows. Section II presents the state-of-the-art on analog IC sizing is presented, providing some insights on machine learning (ML) and some methodologies that are made to estimate applicability to analog IC automation. In Sections III and IV the artificial neural networks (ANNs) are described, presenting its development and training process for two voltage-controlled oscillators (VCOs). Section V presents the results of the implementation of the ANNs in the design of the circuits, where the results are analyzed and discussed. In Section VI the final remarks of the work and a brief discussion on future research directions is addressed.

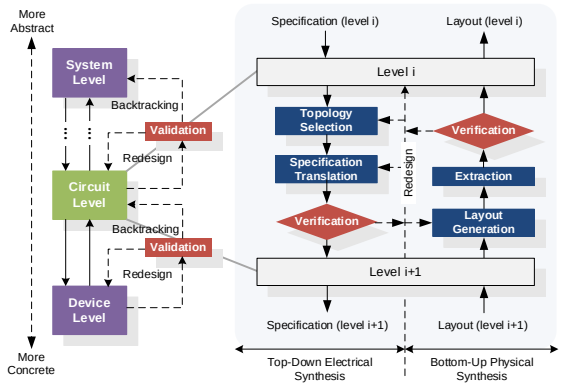

Figure. 1. Steps when designing AMS IC manually. Reprinted from [4].

II. Related Work & Contributions

For many years, humans have been trying to develop machines to help them in daily tasks, to reduce their workload, and to achieve better results. As the advances in technologies kept increasing, the need to develop more complex and automated machines kept growing. This led to the birth of ML. ML is an area of artificial intelligence (AI) that aims at building an expert system, focusing on the statistical properties of data.

As each simulation made within an optimization-based loop is a time-consuming process, there has been a development in techniques to reduce the workload of the simulator. Particularly, ANN models have been extremely popular when addressing this problem.

ANN, or simply called neural network, is an algorithm based on the way the human brain analyses and processes information. ANN consist of node layers, one input layer, one output layer, and at least one, hidden layer. Each layer connects to the next one through links with its associated weight and threshold. ANNs can build effective end-to-end ML systems, being an exceptionally flexible construct. Learning methods have been derived from ANNs such as Deep learning, which is extremely useful when a lot of data is available.

In the work [5], presented in 2003, a neural network-based methodology is used to create fast and efficient models for estimating the performance parameters of Complementary Metal-Oxide-Semiconductor (CMOS) operational amplifier topologies. The results of both efficiency and accuracy of the obtained performance models were demonstrated in a generic algorithm-based circuit synthesis system. This tool is based on performance constraints that are defined by the user and aim to optimize a fitness function. The validation of the performance parameters of the synthesized circuits is done with SPICE simulations and then compared to the ones predicted by the neural network models.

In the work presented in [6], it is used deep neural networks (DNNs) to replace SPICE. In addition to a Multiobjective Optimization (MOO), which is commonly used in analog circuits to identify the tradeoffs imposed by the designer specifications by using POFs, is used a Single-Objective Optimization (SOO). The ANN was trained using data obtained in the MOO phase, therefore requiring no additional step for its training. The ANN replaces the simulator in the later phase, in the SOO phase, reducing the performance evaluation time.

The use of ANNs to find device sizing in analog IC is proving up to be a widely accepted approach and its use can learn and speculate circuit sizing when asked for some target specifications [7][8]. In [9] an ANN is developed to give the channel widths of all the transistors in a circuit when given the output specifications by the designer. The training phase data was performed with different SPICE parameters from the ones used in the test data to show the ability to give the transistor sizes of a circuit for new unknown technology, having no dependency on the SPICE parameters. As a method of validation, two circuits were used, current mirrors and a CMOS differential amplifier. For the first one, a general regression neural network was used, and the results showed that it can estimate the current mirror circuits transistor sizes for never seen technologies, having a 94% accuracy. For the second one, a multilayer perceptron and the results had an accuracy of 90%.

In [10] to produce the sizing for a low noise amplifier, several ANNs are put in sequential order, having as input the intended performance. The results have shown good prediction accuracy, however, the train and tune of such a model have proven to be exceedingly difficult. Having only used 277 handmade sizing solutions for the training phase, this one still needed an outer loop to acquire the model's hyperparameters, which reflected in a train of over 5 hours on such a short dataset. In [11] the sizing for an amplifier is also predicted using ANNs when given its specifications. However, in this one, the model and training phases are different since the test was only performed on 10 samples from the original dataset, and there is not made evaluation on the performance and usability of the model for unknown target specifications.

In the work [12], presented in 2019, a feed-forward time-delay ANN is used to address one of the biggest concerns in today's SoC design and verification, power consumption. Analysing power consumption is still an extremely hard task because of the dependency on time-consuming low-level simulations. The implementation of the ANN comes to increase the functional models of AMS blocks regarding information about their transient power consumption, where this one is trained and then translated into a behavioral modelling language that is tolerant to industrial circuit simulators. All of this, without the need to have manual interaction. To validate this approach, a low power relaxation oscillator was used, and the results confirm that the energy consumption has come close to the considered time range, having an error of only 2.7%, while the needed simulation time has diminished.

As seen, ML and ANNs have been successfully used in the field of analog IC design automation. The Table 1 sums up all the contributions and applications discussed along this Section.

While having the simulator as the evaluation engine is beneficial in both generality and accuracy, it suffers from demanding execution times. Researchers have been using ML to reduce the workload of the simulator, and both SVM and ANNs have shown to be very useful when optimizing simulation-based techniques. Thus, this makes the choice to use ANNs in this work, with the intent to enhance the optimization-based sizing, in an attempt to complement the circuit simulator used.

Table 1 – Summary of Related Work Techniques

Reference	Application	Method
[4], 2003	Simulation-based Enhanced by ANN	ANN
[6], 2020	Simulation-based Enhanced by ANN	DNN
[7][8], 2018 - 2019	Predicting devices size	DL and ANN
[9], 2008	Predicting devices size	ANN
[10], 2015	Predicting devices size	ANN
[11], 2017	Predicting devices size	ANN
[12], 2019	Other Application	TDNN

III. Implementation

The work proposes the development of ANNs, namely DNNs, to enhance the current analog RF IC sizing optimization methodologies, given the sizing of the devices. The goal is to reduce the total effort of the evaluation engine, i.e., the simulator, by reducing the number of candidate circuit sizing solutions that are actually simulated. This is accomplished with the use of two ANNs, one classifier, and one regressor. As this work focuses on VCO circuits, the combination of these two will try to predict whether a certain solution can generate all the performances metrics, i.e., if it will "converge", and when this is the case, at what frequency does the circuit oscillates.

The optimization process is described in Figure 2, where the DNNs play the filter role of selecting from the candidate circuits the most likely to be useful to be presented to the simulator.

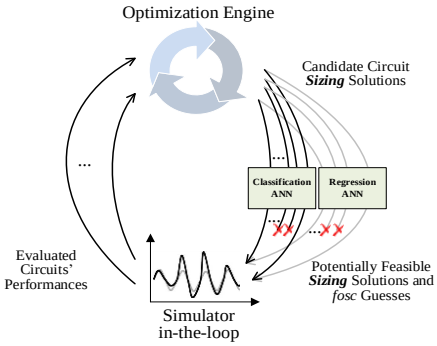

Figure. 2. Proposed Optimization

The development of the ANNs is used, initially, to accelerate the optimization sizing process of a complex dual-mode class C/D VCO described in [13], where a schematic of the circuit is shown in Figure 3.

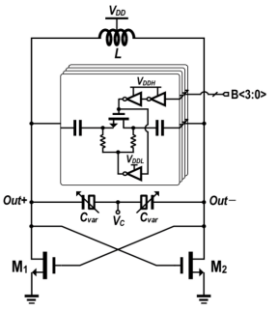

Figure. 3. Dual-mode class C/D VCO schematic reprinted from [13]

This circuit contains 43 devices, and there are 28 optimization/design variables used in [13], such as devices' widths, lengths, and number of fingers, whose sizes must be chosen in order to meet the desired circuit performances. The list of the optimization variables is shown in Table 2.

Table. 2. Optimization Variables

Variable	Units	Min	Grid	Max
Ind_radius	µm	15	5	90
Ind_nturns	-	1	1	6
Ind_spacing	µm	2	1	4
Ind_width	µm	3	1	30
mccl, m1l	nm	60	20	240
mccw, m1w	µm	0.6	0.2	6
mccnf, m1nf	-	1	1	32
mccm	-	1	1	100
moscapw	µm	0.4	0.2	3.2
moscapl	µm	0.2	0.2	3.2
mimvw, mimvl, mim1w	µm	2	0.2	20
r1l, r2l, r3l, r4l	µm	1	0.2	10
r1m, r2m, r3m, r4m	-	1	1	20
nfn1, nfn2, nfp1, nfp2	-	1	1	100

These optimization variables (which impact the sizing of the different devices) will be used as inputs for the ANNs, being these ones evaluated for 9 different tesbenches variations, i.e., typical, and extreme conditions. These different testbenches for extreme conditions are called PVT corners and are outlined in Table 3.

Table. 3. PVT Corners considered

Name	Process	Voltage	Temperature
tt	TT	0.35V	25°C
ff	FF	0.35V	25°C
fs	FS	0.35V	25°C
sf	SF	0.35V	25°C
ss	SS	0.35V	25°C
300mV	TT	0.3V	25°C
400mV	TT	0.4V	25°C
M40dC	TT	0.35V	-40°C
85dC	TT	0.35V	85°C

Each sizing combination is simulated for each PVT corner and, a worst-case tunning range optimization is considered. The circuit is desired to oscillate between 3.9GHz and 4.8GHz, therefore the circuit will be optimized for two tunning modes, b0000 and b1111, for the values of 3.9GHz and 4.8GHz, respectively, generating for each simulation the performances illustrated in Table 4. This will result in 18 simulations, 9 for each tunning mode, where each one produces 10 performances values.

Table. 4. Performances considered

Measure	Units	Description
f_{osc}	GHz	Oscillation frequency
PN@10kHz	dBc/Hz	Phase noise at 10kHz
PN@100kHz	dBc/Hz	Phase noise at 100kHz
PN@1MHz	dBc/Hz	Phase noise at 1MHz
PN@10MHz	dBc/Hz	Phase noise at 10MHz
Power	mW	Power consumption
FOM@10kHz	dBc/Hz	Figure-of-merit at 10kHz
FOM@100kHz	dBc/Hz	Figure-of-merit at 100kHz
FOM@1MHz	dBc/Hz	Figure-of-merit at 1MHz
FOM@10MHz	dBc/Hz	Figure-of-merit at 10MHz

It is necessary to prepare the dataset for each ANN. The tunning mode must be added to the input features, and for the regression ANN the PVT corner as well. Also, for the classification ANN, there is the need to label each PVT corner, corresponding to a binary classification of "converges" or not. To do this, the 10 performances for each PVT corner will have to be grouped up and according to the values of the performances the group will be labelled as "converge" or not, i.e., if there is at least one value that is equal to 0 or non-defined then it is labelled as "not converges", otherwise it is labelled as "converges". This process of data preparation must be done before training the ANNs.

IV. ANNs Structure and Training

For this work, the two ANNs that were developed have similar architectures, with some minor differences. All ANNs were implemented using Python language with the use of different ML libraries such as TensorFlow [14] and Keras [15]. The code was executed on an Intel® Core™ i5-8600K 6 cores CPU 3.60 GHz with 16 GB of RAM.

A. Class-C/D VCO Classification

The classification receives the device sizings and the tuning mode and will do a classification of "converges" or "not converges", which is the classification of each possible candidate circuit sizing solution, as shown in Figure 4. Each output is classified based on the values of the performances produced, for each corner considered. If the values produced are all different than 0 it is classified as "converge", otherwise as "not converge" and discarded. Obviously, the number of neurons in the input and output layers is determined by the number of features and the type of output pretended. In this work, the number of inputs will be the number of the devices

sizings, plus the tunning mode, and one output for each corner analyzed. The training process led to a model with its tuned hyperparameters described in Table 5 and the loss and accuracy of both training validation shown in Figure 5 and Figure 6. By analyzing the loss graphs as the epochs start increasing, the loss curve starts decreasing achieving its best result on the last epochs. At this point, the training curve starts to get lower than the validation one, hence the need to stop rests there, otherwise the model would start to overfit. As for the accuracy graphs, the same results are observed, as the epochs increase the value gets better, in this case it goes higher, which means that the model learns to predict the status of the corners more correctly.

Table. 5. Summary of the Classification Model

Hyperparameter	Value
Input Layer	1 Layer (30 neurons)
Hidden Layers	200,150 neurons
Output Layer	1 Layer (1 neuron)
Activation Functions	Leaky ReLU and Linear
Optimizer	Adam
Regularizer	Dropout (drop rate = 20%)
Loss Function	MSE
Learning Rate	0.0008
Epochs	100
Batch Size	256
Normalization	Min Max (1,2)

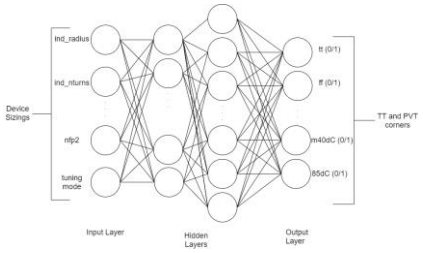

Figure. 4. Classification ANN

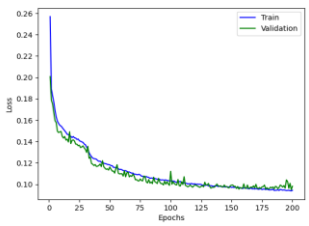

Figure. 5. Training and validation loss

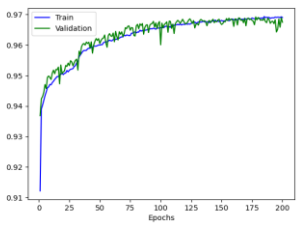

Figure. 6. Training and validation accuracy

B. *Class-C/D VCO Regression*

The regression, besides having the same inputs as the classification, have one additional input, the corner to be considered. The output will determine the oscillatory frequency for that specific corner and tuning mode, as shown in Figure 7.

Having tested and tuned all the hyperparameters, the ANN was trained, and its architecture and parameters are showed in the Table 6. The results of the metrics used for both the training and validation are shown in the Figure 8, Figure 9, and Figure 10. Analyzing the graphs, the model learns to predict the oscillatory frequency, having a lower error value as the epochs keep increasing, up to the point where the model stops training since an overfit would occur if it would continue.

Table. 6. Summary of the Regression Model

Hyperparameter	Value
Input Layer	1 Layer (29 neurons)
Hidden Layers	200,200,250 neurons
Output Layer	1 Layer (9 neurons)
Activation Functions	Sigmoid
Optimizer	Adam
Regularizer	Dropout (drop rate = 20%)
Loss Function	Binary Crossentropy
Learning Rate	0.003
Epochs	200
Batch Size	256
Normalization	Min Max (0,1)

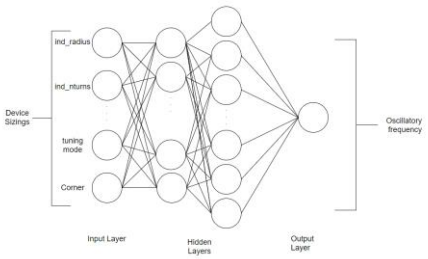

Figure. 7. Regression ANN

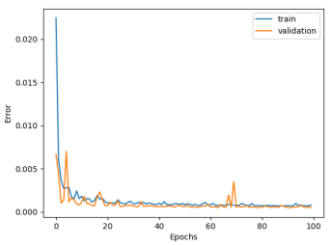

Figure. 8. Training and validation MSE

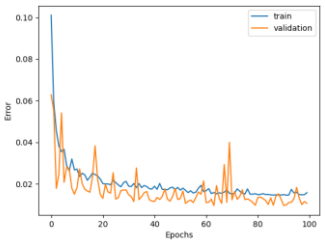

Figure. 9. Training and validation MAE

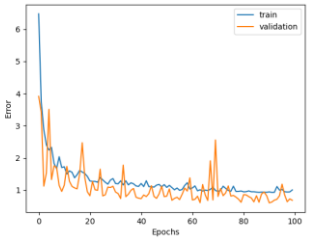

Figure. 10. Training and validation MAPE

C. *Class-B/C Hybrid-Mode VCO*

The results regarding another circuit, Ultralow-Power Complementary Class-B/C Hybrid-Mode VCO, illustrated in Figure 11, are presented, following the logic that was used in the previous section. This circuit was designed for 4.2 to 5.1 GHz Ultralow-Power in a 65-nm CMOS node, having a total of 22 optimization variables, being simulated for 9 different testbenches variations [16]. For this new circuit, the objective is to use the same model as the one used before. Nonetheless, new models have also been trained with its specific hyperparameters for this circuit topology, and the results are compared with the first models for the Class-C/D VCO.

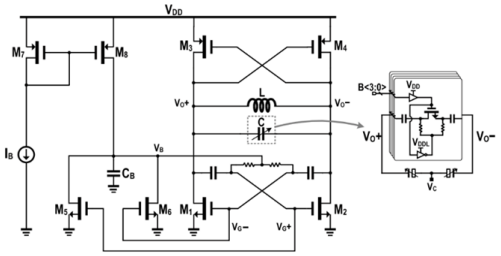

Figure. 11. Ultralow-Power Complementary Class-B/C Hybrid-Mode VCO topology. Reprinted from [16]

The comparison between both models, show that results are marginally different, and so it was proven that the time spent on studying the model for one circuit, does not need to be wasted for a second one, since the model has a high level of generalization, and so is able to present very optimal results when applied to a different VCO.

V. Experimental Results

To test different approaches, a different threshold, used to determine if the points will be presented to the simulator, was tested and the results analyzed. Not only did the threshold came from the usage of the classification ANN, but also from the regression one.

To start the experiments, the threshold was set to 50%, and to make the conditions identical, it was used the same configuration of the optimization without the ANNs, using a population of 256 elements and a total of 150 generations, and the results were obtained with and without the use of the Regression ANN.

A. *50% Threshold*

To start off, the directive to allow the points to be simulated, was a naive approach starting with a 50% threshold, meaning that if the classification ANN predicts, for a given point, that more than half the PVT corners converge, then that point is simulated, otherwise discarded. With this defined, an optimization was conducted, and the points simulated and discarded computed, arriving to the ratio of points simulated and not simulated illustrated in Figure 12. This graph shows that, from the total points that were supposed to be simulated, 18.65% of them were discarded by the ANNs, meaning that almost 20% of the total time required for the optimization was saved with the use of ANNs. Since the optimization without the use of these models lasted 25 days, with the use of these, almost 5 days were economized.

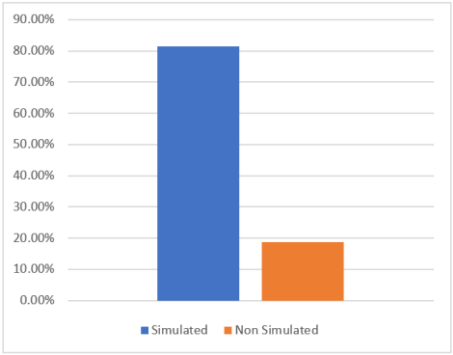

Figure. 12. Ratio of points discarded using a value of 50% threshold

B. *Impact of the Regression ANN*

To prove the value and use of the regression ANN, this one was removed from the implementation, and another optimization with the same characteristics, a population of 256 elements that were optimized for 150 generations, was conducted. This led to a discard of almost 30% of the total points that would be simulated, as 28.23% of the total points were discarded, which was higher than the previous results, as the Figure 13 shows. This increase in points discarded can be explained since, without the regression ANN, the simulator was not able to come up with the best oscillatory frequencies, and so the points that progressed during the evolution of the generations, were not so optimized, hence more solutions were discarded by the classification ANN.

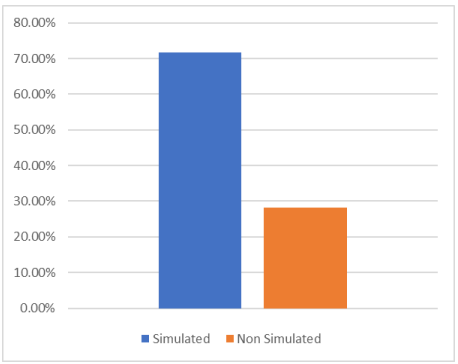

Figure. 13. Ratio of points discarded without the use of the regression ANN

The optimal points for each optimization, with and without the use of the regression ANN, were retrieved and a POF was obtained. These two POFs were then compared against the POF obtained without the use of the ANNs, considered the reference, as depicted in Figure 14. As the results show, the use of the regression ANN proves to be valuable since that, even though its corresponding POF shows that the solutions obtained have better results in terms of phase noise, however, in terms of power the results are considerably worse, never achieving values lower than 1.30E-03. Where with the use of the regression, the POF obtained is very similar to the reference one. Some solutions have worse values of power, however there are some that achieve values lower than 9.00E-04, and better results in terms of phase noise, reaching values of -136.00. Therefore, the use of the regression ANN proves to be essential, as it helps the simulator to converge thanks to the predicted value of the oscillatory frequency.

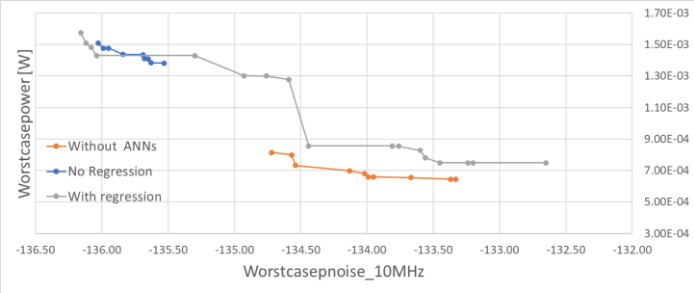

Figure. 14. Comparison of POFs obtained with a value of 50% threshold and with and without the regression ANN versus the original

C. *75% Threshold*

Next, the threshold was defined to 75%, being this value more rigid and so it is expected that the number of points discarded is higher. The same configuration for the optimization was used, and so the points simulated were computed, reaching to the ratio showcased in Figure 15. As expected, the efficiency of the classification ANN was higher, as the points discarded were higher than the results before, reaching a value of 19.65% of points discarded from the total points that would be fed to the simulator.

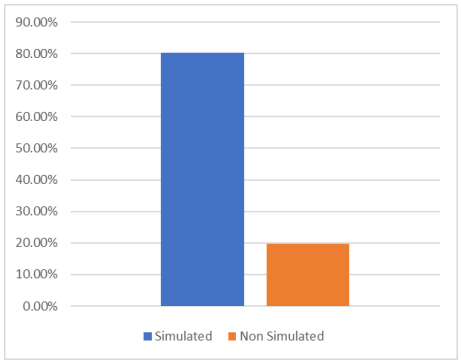

Figure. 15. Ratio of points discarded using a value of 75% threshold

To conclude, the resulting POF was extracted and again compared with the reference one as illustrated in Figure 16. The results show that the points obtained were again very similar to the references one, even reaching values of power and phase noise better than the reference one. In terms of power, all the values are lower than 7.91E-04, and lower than -134.19 in terms of phase noise.

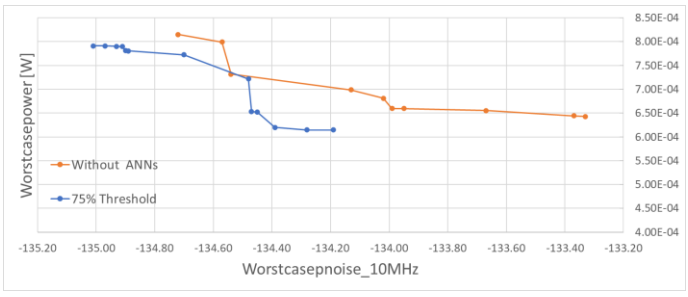

Figure. 16. POF obtained with 75% threshold

To sum up, the results show that the ANNs are accomplishing what is expected, discarding unwanted solutions, being each ANN indispensable. They show apt to perform optimizations in less time, without compromising the results.

D. *New Specifications*

To prove the efficiency of the ANNs with different changes within the same circuit, a new test was made, and the ANNs were faced upon a new setup where the specifications for the circuit were slightly different. For this setup, the circuit was meant to operate at 2.4GHz, having a range of frequency from 2.3GHz to 2.5GHz, and the constraints of the phase noise were changed, being the new ones tighter in 5dBc/Hz.

Having all prepared, a new optimization was executed, using a population of 256 elements for 200 generations, and the ratio of the points simulated and discarded was produced as Figure 17 illustrates. The number of points discarded was low, having a value of only 9.51%, which can indicate immediately that the efficiency of the ANNs were poorer.

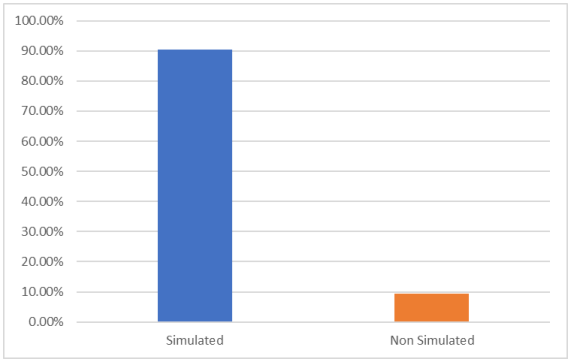

Figure. 17. Ratio of points discarded for the new specifications

Even with a low number of points discarded, the optimal points obtained could present good values, and so the POF was retrieved and compared with the one obtained without the use of the ANNs. Analyzing the results, one can observe that the POF obtained while using the ANNs has better results in term of phase noise, as the solutions are always lower than -138.50. However, that's not the case when looking at power. The values are higher, achieving results that don't get lower than 6.00E-04, while the ones without using the ANNs, are lower, where the values of power never reach 5.00E-04, as observed in Figure 18.

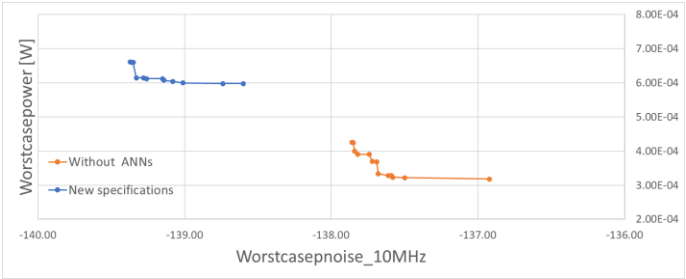

Figure. 18. POF obtained with the new specifications

Ultimately, the results are acceptable, with a reduction of almost 10% of the time, which corresponds to a reduction of almost 3 days since the optimization took about 26 days to complete without the use of ANNs. Even though, the POFs are slightly distanced, the optimal points retrieved proved to be better regarding the phase noise, with slight inferior results in

power. This can lead to the deduction that, the ANNs do perform well when facing different specifications, and so, is possible to use already trained ANNs for different configurations for the circuit that they were trained for.

E. Ultralow-Power Complementary Class-B/C Hybrid-Mode VCO

Finally, the ANNs were implemented in the second circuit, the Class-B/C VCO. The same approach was used as the one used for the Class-C/D VCO, and so it was performed an optimization and the number of points that were simulated were extracted. For this circuit, the configurations were changed, as the ones used without using the ANNs were also modified, and so, a generation of 256 points was used, and it was performed an optimization for 97 generations. As Figure 19 shows, from the total points that were simulated, 17.27% were discarded, being this number close to the 19.65% that was obtained for the first circuit, the Class-C/D VCO, with a value of also 75% threshold. Since the optimization without using ANNs took 25 days, by using these ones, more than 4 days were economized.

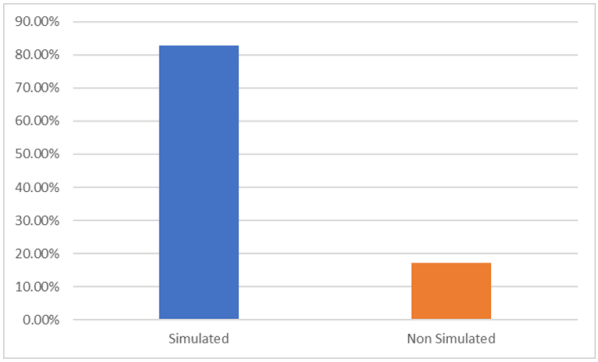

Figure. 19. Ratio of points discarded for the new circuit

To prove the validity of the results, the POF was obtained, and it was compared against the one without the use of ANNs, named reference, as showed in Figure 20. The results show to be very promising, as the values of power obtained are lower, being close 1.50E-04, as the ones obtained without using the ANNs are closer to 2.00E-04. Regarding phase noise, the results are very similar, even recording three points that reach values lower than –130.55, whereas in the reference POF the results never get lower than this value.

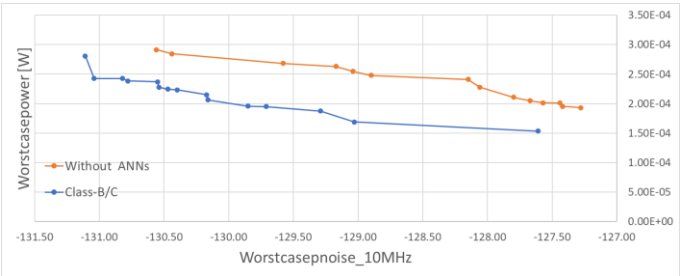

Figure. 20. POF obtained for the Class-B/C

VI. Conclusions & Future Research Directions

This Section exposes not only the conclusions of the work done on this dissertation, but also the aim of the project for further optimizations and use of ANNs applied to analog IC sizing.

The work here exhibited, showed an approach to optimize the sizing of analog IC circuits with the help of two ANNs, one classification and one regression. They both proved to be essential, as the use of the classification and regression ANN discard unwanted solutions, and the use of the last one makes the results even more accurate and approximate to the ones obtained at the first place, therefore reducing the whole optimization process.

Two circuits, with the same constraints, were studied, and its design optimized. The first one was a Class-C/D, where thanks to the ANNs, it was possible to reduce almost 20% of the optimization process, without compromising the results. The optimal points obtained were very similar, even better in some cases, to the ones retrieved without using the ANNs. For the same circuit, the ANNs also proved to work for different specifications, however with less efficiency, as the time saved was close to 10%, and the solutions obtained better in some cases, but worse in other.

A second circuit, an Ultralow-Power Complementary Class-B/C Hybrid-Mode VCO, was analyzed and optimized, having the intention to use the same ANNs used for the previous circuit, the Class-C/D. To do so, a careful study was again made, using the same logic used for the previous circuit, and the new models obtained were discussed and compared with the previous ones. The results were marginally different, proving that, there is no need to spend time training new ANNs, and so emphasizing the generalization of these ones. The design of this circuit, was successful, reducing almost as much time as for the first circuit, and its optimal points showed very promising results.

To sum up, the results proved that the ANNs not only show to be very useful when designing a circuit, but also, for a different one, of the same type, without having the need to do an additional train of the ANNs. However, the results for a different specification fall somehow short of the ones expected, as the model implemented is not able to achieve its best results.

This work proved that the use of ANNs reduce the time to design an analog IC, even if the effort required for their training must be taken in consideration. Having this thought in mind, it is possible to take a step further.

First of all, even though the ANNs prove to have a high level of generalization towards different types of VCOs, there is still required time to train these ones. A future objective would be to reduce this time, which would be possible with additional computation resources. In this work, the training of the ANNs, was performed with human interaction, where each hyperparameter was optimized manually. On a future perspective, this process could be optimized by using algorithms finding the best values for the hyperparameters, resulting on an even further reduction in time.

On a second approach, the training of this ANNs was executed using a dataset that was given, so this one was already known and well defined. However, with the ANN architecture well defined, the implementation of the models can be used without the need for the complete dataset at the beginning, and therefore the train of the ANNs would not be done before, but as part and during the optimization process. This would work in a dynamic way, such that the ANNs could be implemented in the optimization loop, and they would be trained during its process, resulting at first, on using only the normal procedure, without the ANNs, and as soon as the accuracy of the ANNs

would be high enough, they would start to be used and therefore to reduce the optimization time.

One challenge that comes with this approach, is the flexibility that the ANNs would have to show when facing small samples of data and be trained fast and accurate enough for the process to be worth its implementation.

VII. References

[1] G. G. E. Gielen and R. A. Rutenbar, "Computer-aided design of analog and mixed-signal integrated circuits," in Proceedings of the IEEE, vol. 88, no. 12, pp. 1825-1854, Dec. 2000, doi: 10.1109/5.899053.

[2] H. Graeb, "ITRS 2011 Analog EDA Challenges and Approaches," *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Dresden, 2012, pp. 1150-1155, doi: 10.1109/DATE.2012.6176575.

[3] G. Gielen and R. Rutenbar, "Computer-aided design of analog and mixed-signal integratedcircuits, "Proceedings of the IEEE,vol. 88, no. 12, pp. 1825-1854, December 2000.

[4] G. G. E. Gielen and R. A. Rutenbar, "Computer-aided design of analog and mixed-signal integrated circuits," in *Proceedings of the IEEE*, vol. 88, no. 12, pp. 1825-1854, Dec. 2000.

[5] G. Wolfe and R. Vemuri, "Extraction and use of neural network models in automated synthesis of operational amplifiers," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 22, no. 2, pp. 198-212, Feb. 2003.

[6] T. O. Çakıcı, G. İslamoğlu, Ş. N. Güzelhan, E. Afacan and G. Dündar, "Improving POF Quality in Multi Objective Optimization of Analog ICs via Deep Learning," 2020 European Conference on Circuit Theory and Design (ECCTD), Sofia, Bulgaria, 2020, pp. 1-4.

[7] N. Lourenço et al., "On the Exploration of Promising Analog IC Designs via Artificial Neural Networks," 2018 15th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), Prague, 2018, pp. 133-136.

[8] N. Lourenço and et. al., "Using Polynomial Regression and Artificial Neural Networks for Reusable Analog IC Sizing," in 16th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD). IEEE, 2019, pp. 13–16.

[9] N. Kahraman and T. Yildirim, "Technology independent circuit sizing for fundamental analog circuits using artificial neural networks," 2008 Ph.D. Research in Microelectronics and Electronics, Istanbul, 2008, pp. 1-4.

[10] E. Dumesnil, F. Nabki and M. Boukadoum, "RF-LNA circuit synthesis using an array of artificial neural networks with constrained inputs," *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, Lisbon, 2015, pp. 573-576.

[11] N. Takai and M. Fukuda, "Prediction of element values of OPAmp for required specifications utilizing deep learning," *2017 International Symposium on Electronics and Smart Devices (ISESD)*, Yogyakarta, 2017, pp. 300-303.

[12] M. Grabmann, F. Feldhoff and G. Gläser, "Power to the Model: Generating Energy-Aware Mixed-Signal Models using Machine Learning," 2019 16th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), Lausanne, Switzerland, 2019, pp. 5-8.

[13] R. Martins, N. Lourenço, N. Horta, J. Yin, P. Mak and R. P. Martins, "Many-Objective Sizing Optimization of a Class-C/D VCO for Ultralow-Power IoT and Ultralow-Phase-Noise Cellular Applications," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 27, no. 1, pp. 69-82, Jan. 2019, doi: 10.1109/TVLSI.2018.2872410.

[14] F. Ertam and G. Aydın, "Data classification with deep learning using Tensorflow," 2017 International Conference on Computer Science and Engineering (UBMK), 2017, pp. 755-758, doi: 10.1109/UBMK.2017.8093521.

[15] H. Lee and J. Song, "Introduction to convolutional neural network using Keras; an understanding from a statistician," Communications for Statistical Applications and Methods, vol. 26, no. 6, pp. 591–610, Nov. 2019.

[16] R. Martins, N. Lourenço, N. Horta, S. Zhong, J. Yin, P.-I. Mak, and R. P. Martins, "Design of a 4.2-to-5.1 GHz Ultralow-Power Complementary Class-B/C Hybrid-Mode VCO in 65-nm CMOS Fully Supported by EDA Tools," in IEEE Transactions on Circuits and Systems I: Regular Papers (IEEE TCAS-I), vol. 67, no. 11, pp. 3965-3977, Nov. 2020. DOI: 10.1109/TCSI.2020.3009857.