Variation-aware Analog Circuit Sizing with Classifier Chains

Zhengfeng Wu
Department of Electrical and Computer Engineering
Drexel University
Philadelphia, PA
jeff.wu@drexel.edu

Ioannis Savidis
Department of Electrical and Computer Engineering
Drexel University
Philadelphia, PA
isavidis@coe.drexel.edu

***Abstract*—In this work, a simulation-based optimization framework is proposed that determines the sizing of components of an analog circuit to meet target design specifications while also satisfying the robustness specifications set by the designer. The robustness is guaranteed by setting a limit on the standard deviations of the variations in the performance parameters of a circuit across all process and temperature corners of interest. Classifier chains are utilized that, in addition to modeling the relationship between inputs and outputs, learn the relationships among output labels. Additional design knowledge is inferred from the optimal ordering of the classifier chain. A case study is provided, where an LNA is designed in a 65 nm fabrication process. The corners of interest include the combination of the three temperatures of 20°C, 80°C, and 120°C, and the five process corners of typical-typical, slow-slow, fast-fast, slow-fast, and fast-slow. The adoption of classifier chains and the ensemble of classifier chains provides an improvement in the prediction accuracy as compared to the utilization of binary relevance. A qualified design solution is generated that satisfies both the performance and robustness specifications within 5 executed iterations of the design loop.**

I. Introduction & Background

The automation of analog circuit design has drawn particular interest among the research community. The synthesis flow of an analog circuit consists of topology selection, component sizing, and physical design. The sizing of components, which includes both passive and active devices, is a critical step that ensures the selected circuit topology satisfies the target specifications. More recently, machine learning is explored as a means to facilitate the optimization of the sizing of an analog circuit [1] [2].

The goal of applying machine learning is to learn and develop models to map from the design space to the performance space. A multi-label regression or classification problem is formulated, which is then solved by optimization algorithms [3]. In [4], classifiers are applied to predict whether a design point satisfies the provided specifications. However, the interdependence among output labels has not been fully explored or utilized. The prediction models trained for the sizing of an analog circuit are improved by accounting for the relationships among the circuit performance metrics.

Another challenge for the automation of analog circuit design is the proper consideration of the effects that variations have on the output performances. Analog integrated circuits are sensitive to both inter-chip variations introduced by the fabrication process and intra-chip variations resulting from the discrepancy among parameters of individual transistors, such as deviations in the oxide thickness or the threshold voltage. In addition, during circuit operation, environmental effects including changes in temperature result in deviations in the performance of a circuit. With the variations in circuit parameters resulting in yield loss or improper operation, compensating for the effects of the variations increases the design complexity.

978-1-6654-3166-8/21/$31.00 ©2021 IEEE

Therefore, design methodologies must account for the effects caused by circuit variations, while limiting any increase in design complexity. In past literature, multiple variation-aware analog circuit sizing frameworks have been proposed [5], [6]. Direct optimization methods [7] target maximizing the yield characterized by Monte Carlo analysis, which is an effective approach to simulate and model the effects due to variations. Probability density functions are generated from the density estimates of the Monte Carlo samples. However, the use of the Monte Carlo method is computationally expensive. Corner analysis is another approach for the characterization of the effects on circuit performance due to variations. In digital circuits, corner analysis is applied specifically to account for the effects of variations on timing and power consumption. Typically, in addition to temperature and voltage, five process corners are considered: typical-typical (*TT*), fast-fast (*FF*), slow-slow (*SS*), slow-fast (*SF*), and fast-slow (*FS*). The utilization of electronic design automation (EDA) tools such as Cadence allows for the numerical simulation of circuit performances at different process corners and temperatures. Design solutions are considered robust if the specifications are satisfied for all corner cases, or fall within a certain standard deviation from the specifications for all corner cases [5]. Designing for the worst case guarantees robustness but usually results in over-design or renders the problem infeasible to implement. Another approach is design centering [8], which selects design solutions that are farthest from the specification boundaries so that process and environmental variations are tolerated.

In this work, a simulation-based optimization framework is proposed that sizes analog circuit components to meet the design specifications while constraining the variations in the performance of the circuit across all corners of interest within a set bound. Classifier chains are utilized that represent the relationships among output parameters to improve the model accuracy and to provide additional design insight after the completion of the automated sizing methodology.

The remainder of the paper is organized as follows. In Section II, the variation-aware circuit sizing methodology that utilizes classifier chains is described. The results from the execution of the design flow on the sizing of an LNA are provided in Section III. A discussion of the design methodology and the resulting circuit outcomes is provided in Section IV. Finally, some concluding remarks are offered in Section V.

II. Proposed Methodology

For the component sizing of an analog circuit, the design space is denoted as $X \subseteq \mathbb{R}^d$ and the performance space as $Y \subseteq \mathbb{R}^k$. Assume an initial dataset $U = ((x(1), y(1)), ..., (x(n), y(n))) \in (X \times Y)^n$ is randomly sampled from the design space. Binary labels are then assigned to each selected data point of each circuit performance metric based on whether a target threshold is met. The labeled space is denoted as $\hat{Y} \subseteq \{\pm 1\}^k$. One classifier is trained to map h_k: $X \rightarrow \hat{Y}_k$ for the k^{th} circuit performance metric. Therefore, a multi-label classification problem is formulated. In [4], a technique that utilizes adaptive labeling thresholds to train the classifiers is applied. The procedure to determine the set of performance metric thresholds *T* from the provided specification set *S* is given by Algorithm 1, while the routine to assign the labels for training is given by Algorithm 2. The tree-based XG-Boost [9] algorithm is utilized for classification.

Algorithm 1: Adaptively Set the Labeling Thresholds

```
for i = 1 to k do
    if s_i is a lower bound for the i^th circuit
     performance metric y_i then
        t_i ← ε^th percentile of y_i in U;
        if t_i > s_i then t_i ← s_i;
    else
        t_i ← (100 − ε)^th percentile of y_i in U;
        if t_i < s_i then t_i ← s_i;
    end
end
```

Algorithm 2: Assign Labels for Classifier Training

```
for i = 1 to n do
    for j = 1 to k do
        if s_j is a lower bound for the j^th circuit
         performance metric y_j then
            if y_j(i) ≥ t_j then
                ŷ_j(i) = +1;
            else
                ŷ_j(i) = −1;
            end
        else
            if y_j(i) ≤ t_j then
                ŷ_j(i) = +1;
            else
                ŷ_j(i) = −1;
            end
        end
    end
end
```

A. Variation-aware Circuit Sizing

When considering the effects of variations on circuit performance, simulations for each design point are acquired at each corner of interest. The standard deviations of the performance variations across all of the corners for each design point are then calculated. Design points with performance fluctuations that fall below the set threshold T_{thre} of the standard deviation are assigned with positive labels, while all other points are assigned negative labels. The pseudo-code to set the robustness labels is given by Algorithm 3. Therefore, for a set of target performances of dimension *k*, a total of *2k* classifiers are trained with *k* performance predictions and *k* robustness predictions.

Algorithm 3: Assign Labels for Robustness Training

```
for i = 1 to n do
    for j = 1 to k do
        if σ_ij ≤ t_thre_j then
            ŷ_j(i) = +1;
        else
            ŷ_j(i) = −1;
        end
    end
end
```

B. Classifier chains

For a multi-label classification problem, the traditional approach is to train one binary classifier for each labeled target performance metric as shown in Fig. 1, which is known as the binary relevance (*BR*) approach [10]. However, *BR* is based on the assumption that output labels are independent of each other. Research has shown that leveraging the relationships among output labels improves the generalization and application of the trained models [11].

In this work, classifier chains [12] are adopted to model the interdependencies among the outputs. A representation of a classifier chain is shown in Fig. 2. With classifier chains, a set of binary classifiers are combined during the training phase by including output labels from the previous stages as an additional feature set. During the prediction phase, since the true labels are not available, the predictions from the preceding classifier models of the chain are applied instead as features for prediction by the subsequent classifier models in the chain. The procedure to train the classifier chain is described by Algorithm 4. The procedure for prediction with the trained classifier chain is described by Algorithm 5.

With classifier chains, the correlation among target labels is considered. The order in which the classifiers are organized is a key parameter that affects the performance of the model. The first model in the chain is a classifier trained on the original input features, while additional output features are included as training features in the remaining models of the chain.

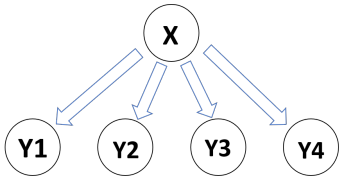


Fig. 1. Diagram depicting a multi-label classification based on the binary relevance approach [10].

C. Ensemble of Classifier Chains (ECC)

An ensemble improves the prediction accuracy of a model and reduces model overfitting [12]. When considering an ensemble of classifier chains, a total of *m* chains are trained, which are denoted as $C_1, C_2, ..., C_m$. If enumeration of all

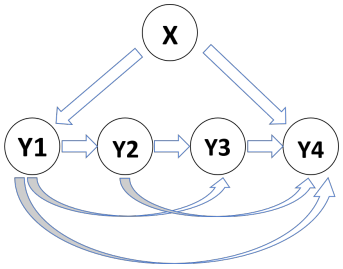


Fig. 2. Diagram depicting a classifier chain that maps from the input feature space to a four-dimensional output space [12].

Algorithm 4: Training of a Classifier Chain

Training set $U = (x(1), \hat{y(1)}), ...(x(n), \hat{y(n)})$;
Order the chain of the k classifiers as 1,2...k;
for $j = 1$ *to* k **do**
  Initialize U' as empty set;
  for $(x, \hat{y}) \in U$ **do**
    $U' = U' \cup ((x, \hat{y_1}, ...\hat{y_{j-1}}), \hat{y_j})$;
    $train\ C_j : U' \rightarrow \hat{y_j} \in \{0, 1\}$;
  end
end

Algorithm 5: Prediction with a Classifier Chain

Initialize Y as empty set;
for $j = 1$ *to* k **do**
  $Y = Y \cup (\hat{y_j} \leftarrow C_j : (x, y_1, y_2, ...y_{j-1}))$;
end
return Y as prediction

the possible orders of the chain is possible, then $m = k!$ classifier chains are trained, where k is the number of output labels. When $m < k!$, each classifier chain in the ensemble is trained with a random ordering of individual classifiers. The predictions are summed and averaged for each label. A threshold is utilized to determine the final predicted labels. The pseudo-code for prediction by the ensemble of classifier chains is given by Algorithm 6.

Algorithm 6: Prediction with Ensemble of Classifier Chains

Train m classifier chains;
for *j=1 to k* **do**
  for *i=1 to m* **do**
    Predict y_j with the i_{th} classifier chain;
  end
  y_{final_j}=sum(y_j)/m;
  Apply decision threshold to get $\hat{y}_{final_j}$;
end
return Y=($\hat{y}_{final_1}$,..., $\hat{y}_{final_k}$) as prediction

D. Design Flow for Variation-aware Component Sizing with Classifier Chains

The classifiers for predicting the circuit parameters are trained on a 'default' standard corner, which was set to *TT* at 20°C in this work. The classifiers for predicting robustness are trained with the standard deviations calculated for all corners (*e.g.,* process, voltage, and temperature) and for each design point (selected component sizes).

After the training of the ensemble of classifier chains, for each iteration of the design loop shown in Fig. 3, a multi-objective search is executed to determine points that simultaneously maximize the predicted probability scores for all of the specifications, as given by

$$x^* \in arg\max(p_1(x), ..., p_k(x), r_1(x), ..., r_k(x)), \quad (1)$$

where $p_k(x)$ is the probability score to predict whether the k^{th} circuit performance parameter satisfies the target specification, and $r_k(x)$ is the probability score to predict whether robustness is satisfied for the k^{th} circuit performance parameter. Both the $p_k(x)$ and $r_k(x)$ scores are outputs from the classifier chain. SPICE simulations are then executed to verify the candidate solutions.

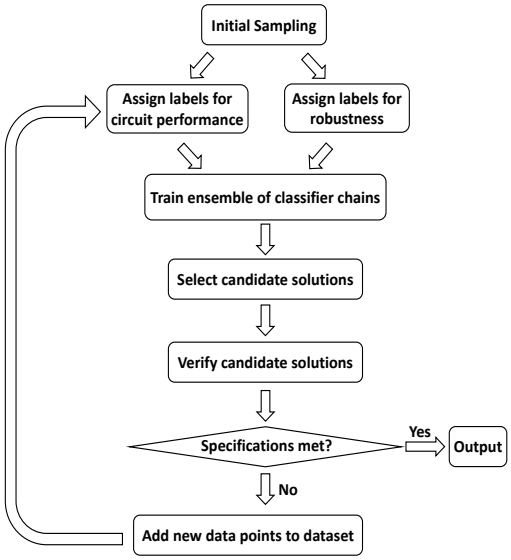


Fig. 3. Proposed design flow that utilizes classifier chains for the sizing of an analog circuit.

III. Simulation Results

The proposed framework is applied to the design of a differential low-noise amplifier (LNA), which is shown in Fig. 4, in a 65 nm technology. The design set consists of nine variables: the sizes of inductors L_{g1}, L_{d1}, and L_{s1}, the widths of transistors M_1, M_3, and M_5, the size of capacitor C_{g1}, and the biasing voltages V_{b1} and V_{b3}. The sizes of the remaining components are set based on symmetry. The transistor length is set to the minimum of 65 nm. The target performance metrics include the *gain*, *noise figure* (*NF*), *third-order intercept point* (*IP3*), and *power consumption*. The target design variables are constrained to

$$\begin{cases} 60\ nm \leq transistor\ widths \leq 900\ \mu m, \\ 0.01\ nH \leq inductor\ sizes \leq 12\ nH, \\ 30\ fF \leq capacitor\ sizes \leq 20\ pF, \text{ and} \\ 0\ V \leq biasing\ voltages \leq 1.2\ V. \end{cases} \quad (2)$$

The specifications for the performance metrics of the circuit are given as

$$\begin{cases} \textit{Gain} \geq 10\ dB, \\ \textit{NF} \leq 2.8\ dB, \\ \textit{IP3} \geq -5\ dBm, \text{ and} \\ \textit{Power} \leq 20\ mW, \end{cases} \tag{3}$$

and the robust requirements of the circuit are given as

$$\begin{cases} \sigma_{Gain} \leq 1\ dB, \\ \sigma_{NF} \leq 0.5\ dB, \\ \sigma_{IP3} \leq 1\ dBm, \text{ and} \\ \sigma_{Power} \leq 5\ mW. \end{cases} \tag{4}$$

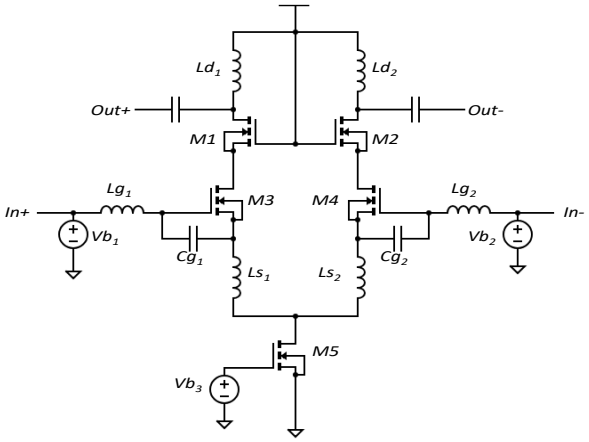
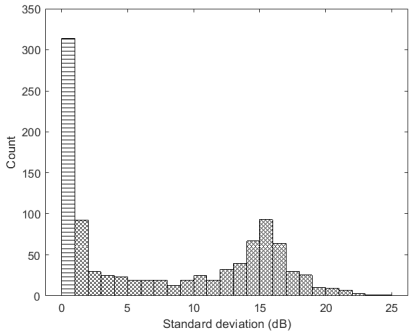


Fig. 4. Circuit schematic of a differential low noise amplifier.

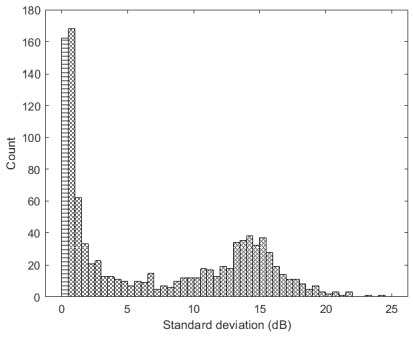
The initial dataset contains 1000 design points. 15 corner cases are simulated for each design point, which are given by the combination of the three temperatures of 20°C, 80°C, and 120°C, and the five process corners of *TT*, *FF*, *SS*, *SF*, and *FS*. The *TT* process corner at 20°C is considered as the default standard case, while the remaining corner cases are acquired primarily for characterizing the variations in the performance of the circuit. For the initial acquired dataset, the distribution of standard deviations of the variations in the performance across the 15 corner cases is shown in Fig. 5. The striped bars indicate the design points that satisfy the user-specified robustness criterion for each performance specification, while the cross bars indicate the design points with performance variations that exceed the set thresholds. Without accounting for robustness, the generated design solutions potentially result in large variations in the circuit performance parameters at different corners (falling to the right side of each plot in Fig. 5).

After the initial dataset is acquired, the design flow is executed with the adaptive labeling threshold method applied to the performance specifications. The ϵ term, as defined in Algorithm 1, is set to 95. For the prediction of the robustness of the circuit, as the initial dataset contains sufficient data points with both positive and negative labels, the labeling thresholds on the standard deviations that constrain the variation in the performance parameters are set directly to the target values.

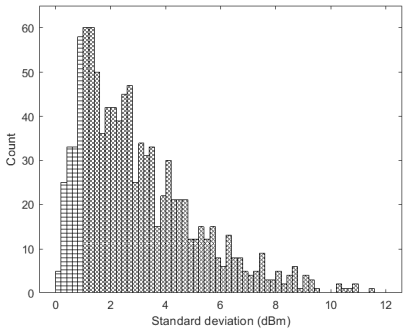
Classifier chains are first trained only on data from the standard corner case without considering variations. For the four circuit performance metrics, there are 4! (24) combinations of possible orderings of the chain. Classifier chains with all 24 possible combinations are trained. The results from the characterization of the performance of the models are shown in Fig. 6.



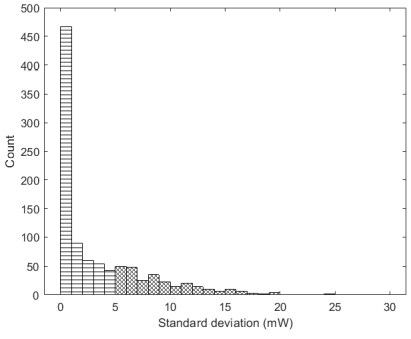
(a)



(b)



(c)



(d)

Fig. 5. Distribution of standard deviations of the performance variations across the 15 corner cases for the initial 1000 data points acquired for a) *gain*, b) *NF*, c) *IP3*, and d) *power consumption*.

From the results shown in Fig. 6, the best six classifier

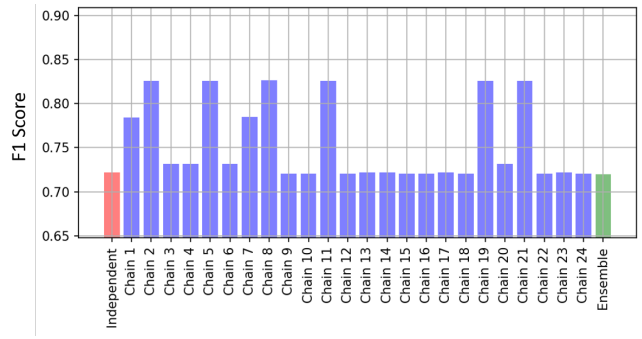


Fig. 6. Average F1-scores for the standalone classifiers based on binary relevance (red), all 24 possible orders of classifier chains (blue) of the four performance parameters, and the ensemble of classifier chains (green) for the prediction of the four circuit performance parameters for the standard typical-typical corner at 20°C.

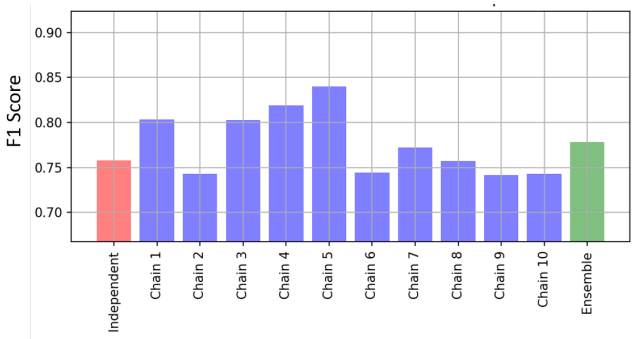


Fig. 7. Average F1-scores for the standalone classifiers based on binary relevance (red), 10 random orders of classifier chains (blue), and the ensemble of the 10 random orders of classifier chains (green) for the prediction of the eight target performance parameters, where four circuit specification parameters and four robustness parameters are predicted.

chains produce F1 scores that outperform the independent models trained based on the binary relevance method by more than 0.1 points. The orders of the six chains are listed in Table I, with 0 representing *power*, 1 representing *IP3*, 2 representing *NF*, and 3 representing *gain*.

TABLE I
ORDERING OF THE SIX CLASSIFIER CHAINS WITH THE HIGHEST F1 SCORES, WHERE 0 REPRESENTS *power*, 1 REPRESENTS *IP3*, 2 REPRESENTS *NF*, AND 3 REPRESENTS *gain*.

Chain 2	$0 \rightarrow 1 \rightarrow 3 \rightarrow 2$
Chain 5	$0 \rightarrow 3 \rightarrow 1 \rightarrow 2$
Chain 8	$1 \rightarrow 0 \rightarrow 3 \rightarrow 2$
Chain 11	$1 \rightarrow 3 \rightarrow 0 \rightarrow 2$
Chain 19	$3 \rightarrow 0 \rightarrow 1 \rightarrow 2$
Chain 21	$3 \rightarrow 1 \rightarrow 0 \rightarrow 2$

Among the six optimal classifier chains, the prediction of the noise figure is always placed last in the chain, which indicates that the prediction of the noise figure is the least accurate and provides the least information to the training of the other classifiers. The classifier chain provides higher performance when models with the highest confidence are placed first, while inaccurate models are placed near the end of the chain so that any error is minimally propagated along the chain.

Including an additional four classifiers to account for the prediction of the robustness of the circuit to variation, the chain now consists of eight classifiers. As the enumeration of 8! combinations to identify the optimal chain sequence is computationally expensive, 10 random combinations of the eight classifiers are trained. Execution of Algorithm 6 provides a prediction of the robustness of the circuit with the ensemble of 10 random classifier chains. The results of the performance of the model are shown in Fig. 7. The results indicate that the ensemble outperforms the trained standalone classifiers.

The design flow, as depicted in Fig. 3, is then executed based on the ensemble of the 10 random classifier chains. A summary of the results for the design of the LNA is provided in Table II. A qualified design solution is generated that satisfies both the specification and robustness requirements after five iterations of the training/inference loop of the ensemble of classifier chains. The total number of design points simulated is 1462, with 1000 points for the initial dataset and 462 points generated while executing the design loop, where an average of 92 candidate solutions are generated each iteration. Each design point requires 15 simulations to account for all corner cases.

TABLE II
SUMMARY OF THE RESULTS FROM THE DESIGN OF THE LNA USING THE ENSEMBLE OF 10 RANDOM CLASSIFIER CHAINS.

Parameters	**Generated Designs**
M_1=M_2	290.5 µm
M_3=M_4	144.8 µm
M_5	137.6 µm
L_{g1}=L_{g2}	3.40 nH
L_{d1}=L_{d2}	3.98 nH
L_{s1}=L_{s2}	0.839 nH
C_{g1}=C_{g2}	0.565 pF
$V_{b1} = V_{b2}$	0.770 V
V_{b3}	0.957 V
gain	10.76 dB
NF	2.78 dB
IP3	-2.65 dBm
power	16.8 mW
σ_{gain}	0.74 dB
σ_{NF}	0.45 dB
σ_{IP3}	0.47 dBm
σ_{power}	4.9 mW
Num. of iterations	5
Num. of samples	21930 (1462×15)

IV. DISCUSSION

With the proposed methodology, in addition to training four classifiers to predict the four performance specifications of the circuit, four additional constraints are added to predict the variations in the performance parameters across different corner cases of interest. The adoption of classifier chains allows for the modeling of the interdependencies among output labels. Based on the results of the performance of the models shown in Fig. 6 and Fig. 7, some classifier chains boost the F1 score as compared to the baseline of standalone classifiers. In addition, the ensemble of 10 random classifier chains also outperforms the eight standalone models for both performance prediction and robustness prediction.

Two implementations of classifier chains are presented. The first approach is to enumerate all of the possible orders of the chain and adopt the optimal order of classifiers. Execution of the first approach is feasible when the number of output labels is small (i.e., four). When the number of output labels is large, the number of possible order combinations of a classifier chain increases exponentially, which indicates that the second

approach of applying an ensemble of randomly selected chains is a better option. The number of individual classifiers required when either the binary relevance or classifier chain technique is applied scales linearly with the size of the label set. Therefore, there is no overhead in computational resources required to train classifier chains. However, during the optimization phase, each seed requires m times more functional evaluations when the ensemble is utilized as compared to applying only the best chain, where m is the number of classifier chains in the ensemble.

Among the generated candidate solutions, the plot of the standard deviation of the power consumption variation as a function of the power consumption is shown in Fig. 8 when all remaining performance and robustness requirements are satisfied. From the results shown in Fig. 8, the variation in the power consumption across all corners is positively correlated with the amount of power consumed. The results indicate that the performance metric and the robustness metric (standard deviations that characterize the variation) are correlated. Therefore, the modeling of the interdependencies amongst output labels is necessary and provides additional design information as compared to only modeling the mapping from the input to the output space.

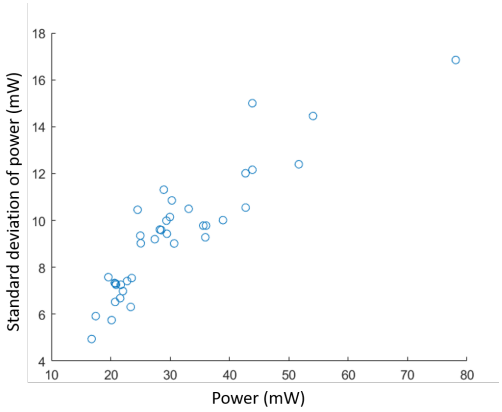


Fig. 8. Relationship between the *power consumption* and the standard deviation of the power consumption variation, σ_{Power}, when all remaining performance and robustness constraints are satisfied.

Corner analysis is based on the assumption of a fixed value in the variation of a physical parameter p_i [13]. The relation between the variation in a physical parameter and the variation in a circuit performance parameter y is given by

$$\Delta y = \sum_i \frac{\partial y}{\partial p_i} \Delta p_i. \tag{5}$$

The sensitivity term, $\frac{\partial y}{\partial p_i}$, varies for each performance metric as the sizes of the components and the bias voltages differ. Therefore, in the ideal case, corner models must be generated for each performance metric separately. Applying fixed variations for p_i regardless of the sensitivities results in inaccuracies in the model. However, the proposed design methodology still applies as the simulation data is considered as ground truth.

An additional alternative to account for the effects of variations on the performance of the circuit is to train classifiers that consider the worst cases of the dataset. The limitation of the approach is that the worst-case performance often occurs in different corner cases for different target circuit specifications. As an example, the worst-case *power consumption* occurs when the *FF* corner is considered at 120°C. However, the worst-case *gain* occurs when the *SS* corner is considered at 20°C. Designing for the worst cases, therefore, results in ambiguous outcomes. Comparatively, designing based on the *TT* corner while constraining all the performance variations across all corners within a certain standard deviation, as proposed in this paper, provides viable solutions.

V. Conclusions

In this work, a simulation-based optimization framework is proposed that determines the sizing of components of an analog circuit to meet target design specifications while also satisfying the robustness specifications set by the designer. The robustness is guaranteed by setting a limit on the standard deviations of the variations in the performance parameters of a circuit across all process and temperature corners of interest. Classifier chains are utilized that, in addition to modeling the relationship between inputs and outputs, learn the relationships amongst output labels.The proposed methodology is applied to the design of an LNA in a 65 *nm* fabrication process. The use of classifier chains and the ensemble of classifier chains provides an improvement in the prediction accuracy as compared to the binary relevance method. A qualified design solution is generated that satisfies both the performance and robustness specifications across all of the corners considered. The *gain*, *noise figure*, *IP3*, and *power consumption* of the design of the LNA are 10.76 *dB*, 2.78 *dB*, -2.65 *dBm*, and 16.8 *mW*, while the standard deviations across all considered corners are 0.74 *dB*, 0.45 *dB*, 0.47 *dBm*, and 4.9 *mW*, respectively.

References

[1] Z. Gao, J. Tao, F. Yang, Y. Su, D. Zhou, and X. Zeng, "Efficient performance trade-off modeling for analog circuit based on Bayesian neural network," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 1–8, Nov. 2019.

[2] H. Wang, K. Wang, J. Yang, L. Shen, N. Sun, H. Lee, and H. Song, "GCN-RL Circuit Designer: Transferable Transistor Sizing with Graph Neural Networks and Reinforcement Learning," *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 1–6, Jul. 2020.

[3] M. Barros, J. Guilherme, and N. Horta, "Analog circuits optimization based on evolutionary computation techniques," *Integration, the VLSI Journal*, Vol. 43, No. 1, pp. 136–155, Jan. 2010.

[4] Z. Wu and I. Savidis, "CALT: Classification with Adaptive Labeling Thresholds for Analog Circuit Sizing," *Proceedings of the 2020 ACM/IEEE Workshop on Machine Learning for CAD*, pp. 49–54, Nov. 2020.

[5] X. Li, P. Gopalakrishnan, Y. Xu, and L. T. Pileggi, "Robust Analog/RF Circuit Design With Projection-Based Performance Modeling," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 26, No. 1, pp. 2–15, Jan. 2007.

[6] T. McConaghy and G. Gielen, "Globally Reliable Variation-Aware Sizing of Analog Integrated Circuits via Response Surfaces and Structural Homotopy," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 28, No. 11, pp. 1627–1640, Oct. 2009.

[7] S. Director, P. Feldmann, and K. Krishna, "Statistical integrated circuit design," *IEEE Journal of Solid-State Circuits*, Vol. 28, No. 3, pp. 193–202, Apr. 1993.

[8] A. Seifi, K. Ponnambalam, and J. Vlach, "A unified approach to statistical design centering of integrated circuits with correlated parameters," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Vol. 46, No. 1, pp. 190–196, Jan. 1999.

[9] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, Aug. 2016.

[10] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, Vol. 37, No. 9, pp. 1757–1771, Mar. 2004.

[11] M. Zhang and Z. Zhou, "A Review on Multi-Label Learning Algorithms," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 26, No. 8, pp. 1819–1837, Aug. 2014.

[12] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier Chains for Multi-label Classification," *Machine Learning*, Vol. 85, No. 3, pp. 254–269, Aug. 2009.

[13] C. C. McAndrew, I. Lim, B. Braswell, and D. Garrity, "Corner models: Inaccurate at best, and it only gets worst," *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 1–4, Sep. 2013.