On the use of Bayesian Networks for Resource-Efficient Self-Calibration of Analog/RF ICs

Martin Andraud*, Laura Galindez*, Yichuan Lu†, Yiorgos Makris†, Marian Verhelst*
*KU Leuven, Dept. of Electrical Engineering (ESAT), MICAS group, Belgium
{martin.andraud, laura.galindez, marian.verhelst}@esat.kuleuven.be
†Department of Electrical Engineering, The University of Texas at Dallas, Richardson, TX 75080, USA
{yichuan.lu,yiorgos.makris}@utdallas.edu

***Abstract*—Over the past few years, several self-calibration methodologies have proven their efficiency to calibrate analog and radio-frequency circuits against process variations. Specifically, statistical techniques based on machine-learning have been proposed to recover yield loss and even enhance circuit performances. In addition, these techniques enable to calibrate circuits after a single performance test, i.e. in one-shot. However, towards fully-integrated calibration techniques, the inference part of the machine learning algorithm needs to be performed as energy-efficiently as possible to reduce calibration cost to a minimum. Following the path of resource-efficient machine learning, this work explores an alternative to state-of-the-art Neural Network based statistical techniques. Specifically, we investigate the opportunities of using Bayesian Networks for resource-efficient on-chip statistical calibration of analog/RF circuits. Results will show that several improvements can be achieved using Bayesian Networks: (a) provide a comprehensive calibration framework with explicit relationships between parameters (b) demonstrate similar prediction accuracies that neural networks (c) optimize across several performance parameters with a single network and in a single query and (d) enable a more energy-efficient hardware implementation. The proposed self-calibration algorithm is applied to a low-noise amplifier fabricated with IBM's 130nm CMOS process, leading to a significant reduction in the number of operations required to obtain the best tuning knob setting.**

I. Introduction

The constant downscaling of CMOS technologies has resulted in a steadily advancing level of integration for mixed-signal circuits and systems. Yet, this extreme miniaturization comes with an increased susceptibility to manufacturing process variations. Regarding analog and Radio Frequency (RF) Integrated Circuits (ICs), this susceptibility becomes a limiting factor in their design. Specifically, maintaining a very low yield loss requires to either 1.) increase the design margins of the circuit, which sacrifices the overall performance; or alternatively 2.) develop methodologies which aim at compensating for these process variations post-manufacturing, while maintaining the desired level of performance. However, implementing such post-manufacturing method for analog/RF ICs is a challenge, since a single industrial test of their performances already comes with a high cost for a typical system-on-chip [1].

A potential solution, recently explored by the state-of-the-art (SotA), is to rely on artificial intelligence to perform statistical post-manufacturing self-calibration [2]–[4]. Essentially, these techniques use machine learning algorithms to estimate process variations through low-cost sensors, and tune internal circuit knobs to compensate for them. Self-calibration using only a single test iteration, called "one-shot", have even recently been presented [5]–[7]. One step further is to fully integrate the calibration procedure on-chip, following up on Built-In Self-Test (BIST) developments [8], [9]. Several benefits can be expected, such as a further reduction of post-manufacturing calibration cost and time. In addition, it offers the possibility for the technique to compensate for more variations emerging in the field, due to, for instance, aging or environmental variations [4]. However, if several performances have to be considered simultaneously, the search for the optimal circuit configuration is usually the result of compromises between all circuit performances, which is generally difficult to be found manually. This requires the use of an optimizer [5], [6], [9], [10]. On-chip, this optimizer is usually embedded in a processor, which can consume a relatively large amount of resources [10].

Following the direction of energy and time-efficient embedded machine learning algorithms, this work investigates the use of Bayesian Networks (BN) to perform the self-calibration of analog and RF ICs. Its main objective is to highlight potential advantages of BNs over SotA works using e.g. Neural Networks (NN). Indeed, the use of BNs for the current application context enables to:

1) Provide a comprehensive calibration framework: given the ability of BN to incorporate expert knowledge and be represented graphically, the calibration problem can be made explicit and be easily adapted to the considered case study.

2) Demonstrate prediction accuracies similar to neural networks: this is enabled by the ability of BN to capture probabilistic relations among all circuit variables.

3) Find best tuning knobs without using an optimizer, even for multi-parameter estimation: given a desired performance and an observed state of the process sensors, the most probable tuning knob setting is obtained in a single network query reducing the tuning hardware overhead. It remains true even when considering several performances explicitly in the BN.

4) Achieve an efficient hardware implementation: we will show that an efficient hardware implementation for probabilistic inference is possible by using Arithmetic Circuits (ACs).

978-1-5386-8382-8/18/$31.00 ©2018 IEEE

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:27:21 UTC from IEEE Xplore. Restrictions apply.

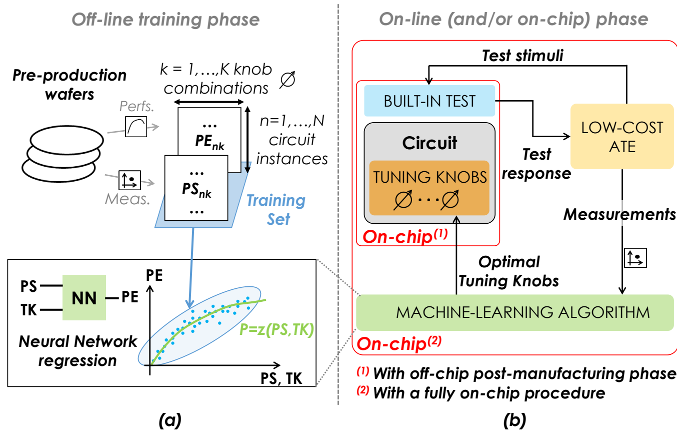

Fig. 1. Illustration of machine-learning based one-shot self-calibration schemes

The remainder of this paper is organized as follows. Section II will detail the related SotA. Section III will introduce how BNs can be exploited for self-calibration. The proposed self-calibration methodology will be detailed in section IV and section V will present results on the considered case study.

II. Previous work on self-calibrated ICs

Any low-cost calibration technique should not rely on direct performance tests executed with an Automatic Test Equipment (ATE). This approach significantly increases test time and cost, which are already pushed to their limits. Solutions to this issue can then be broadly classified in two categories: *direct* [9]–[11] and *statistical* [3]–[7] techniques.

A. Direct self-calibration techniques

Direct methodologies aim to integrate the performance measurement and calibration circuitry on-chip, with the objective to evaluate performance targets directly. Together with on-chip measurement techniques, such as time- or frequency-domain analysis, this approach requires an on-chip optimizer [9]–[11] to find the best multi-performance trade-off. An example of such a fully integrated on-chip analog circuit calibration implementation has been presented in [9]. Although high accuracy calibration can be achieved with this methodology, the on-chip costs are is still high due to the need for additional resources, such as an embedded Digital Signal Processor (DSP). In addition a large number of tests is usually required and the measurement techniques used for analog circuits cannot be applied in a straightforward way for RF circuits, since they need to be tested at higher frequencies. In this work, we will focus on techniques that do not require numerous tests, or the use of an optimizer to limit on-chip resources at maximum.

B. Statistical self-calibration techniques

Statistical self-calibration is based on the alternate test paradigm [2], where only indirect process measurements are used. As these measurements do not directly predict the performance of the circuit, statistical techniques are required to model the relationship between these indirect measurements and the estimated chip performance. Fig.1 shows their key operative elements: 1.) BIST structures, which are low-cost sensors that extract an estimate of process variations. 2.) Tuning knobs, which adjust the performances of the circuit. 3.) A machine-learning block, usually implemented with a NN, which models the relationship between the process sensors PS, the tuning knob settings TK and the chip performance PE; and essentially serves as a regression algorithm to predict the function $z()$ that relates them. This model is pre-trained off-line, with a representative dataset of circuit instances, issued for example from pre-production wafers. Then, at post-manufacturing time, calibration is achieved by: (a) measuring the process sensor values with the BIST structures, (b) predicting the circuit performances through the machine-learning model and (c) determining the best tuning knob settings to achieve the desired performance. This stage can take place off-chip [5], [6] or on-chip [4], [7], [8] as illustrated in fig.1.

To achieve a faster and lower-cost calibration, previous research have shown that the complete procedure can be performed in a unique test iteration, i.e. by measuring process sensors only once. This is referred to as *one-shot* calibration [5]–[7]. This one-shot property can be pursued in several ways. For instance, in [6], it is assumed that process variations and tuning knobs act orthogonally on the performances, which enables to run a composite regression model without taking into account the interactions between tuning knobs and process variations. In [5] such independence is achieved by design, by using non-intrusive built-in sensors that are transparent to the circuit. However, in both cases, if several performances need to be predicted, the technique requires one machine-learning algorithm per performance and a strategy to find the optimal tuning knob setting, i.e. an optimizer. To be fully embedded on-chip, integrating this optimizer leads to the same resource issues as direct methodologies.

Yet, the use of this optimizer can be avoided. A different approach is used in [7], where it is proposed to manually calibrate a set of devices to gather the data necessary to build the regression model, which then predicts directly the best tuning knob setting regarding the sensor values, in a one-shot procedure. However, this technique assumes that each knob control a specific performance parameter independently of the other, which is a desirable attribute but often difficult to achieve in practice during design. In [8] it is proposed to avoid crossing the digital domain and efficiently implement a NN directly in the analog domain. The best tuning knob settings are subsequently found by measuring the process sensors exhaustively going through the tuning knob combinations to find the best one. This approach is more energy-efficient but a limitation is that it only allows for the prediction of a single performance, expressed as a figure of merit (FoM) that trades off circuit performance and power consumption. Yet, calibrating for a given FoM does not guarantee that specifications are individually met for each performance.

Solutions using low on-chip resources, and simplifying the overall calibration process while explicitly considering several performances, are lacking in the current SotA. We propose the use of on-chip BN evaluation to achieve this goal.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:27:21 UTC from IEEE Xplore. Restrictions apply.

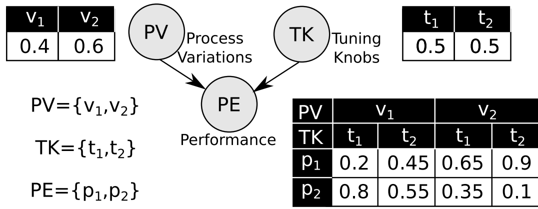

Fig. 2. Illustrative example of a Bayesian Network applied to circuit self-calibration

III. Conceptual application of Bayesian Networks towards circuit self-calibration

Bayesian Networks (BNs) are directed acyclic graphs that compactly encode a joint probability distribution $Pr(F_1, ..., F_n)$ over a set of random variables $\{F_1, ..., F_n\}$ [12]:

$$Pr(F_1, ..., F_n) = \prod_{i=1}^{n} Pr(F_i|\Pi_{F_i}). \quad (1)$$

Within this distribution, Π_{F_i} denotes the parents of F_i and $Pr(F_i|\Pi_{F_i})$ are the conditional dependencies between variables and their parents, which can be represented as Conditional Probability Tables (CPTs). The graphical representation of a BN comprises nodes that represent the random variables and edges that encode the probabilistic dependencies among them.

Fig.2 presents an example of a BN that models the probabilistic relations among variables in a simplistic circuit for a self-calibration application. In this model, the circuit performance PE depends on process variations PV, and tuning knob settings TK, as shown by the edges' directions. The conditional probability distributions among them, which in a real application can be learned from training data, are represented as CPTs. As such, this BN captures all knowledge and dependencies necessary to enable the calibration of the circuit. Specifically, such calibration would consist of observing process variations PV, and inferring the required tuning knob setting TK which would result in the desired performance PE under the observed circumstances. We will pursue this by using the capability of BNs to infer the probability distribution of variables of interest $\mathbf{Q}$ when evidence $\mathbf{e}$ is available from observations ($Pr(\mathbf{Q}|\mathbf{e})$).
Practically, this consists of following steps: 1.) We observe the process variations through indirect sensors. In this example we can e.g. assume an observation of PV=v_1. 2.) Then, we target a certain post-calibration performance, e.g. $PE = p_2$. Both observations will together describe the *evidence* $\mathbf{e_0}$ of the BN query, in the current example $\mathbf{e_0}$=$\{PV$=v_1, PE=$p_2\}$. 3.) Now we want to find the most likely tuning knob setting TK to meet the aforementioned conditions by querying the network for $Pr(TK|\mathbf{E})$ for all possible values of TK. In our example, one can see that $Pr(TK$=$t_1|\mathbf{E}$=$\mathbf{e_0}) = 0.8$ and $Pr(TK$=$t_2|\mathbf{E}$=$\mathbf{e_0}) = 0.55$. Here, TK=t_1 is more likely to fulfill the condition and could thus be selected.

Yet, such process would be very expensive in practice, as it still requires to exhaustively query for all possible tuning knob combinations. In this paper, we achieve the full calibration procedure in a single step by exploiting a search technique for the *Most Probable explanation* (MPE) which is formally defined as a complete variable instantiation that is consistent with evidence $\mathbf{e}$ and has the highest probability [12]:

$$MPE(\mathbf{e}) = arg \max_{\mathbf{x} \sim \mathbf{e}} Pr(\mathbf{x}) \quad (2)$$

Going back to the example where $\mathbf{e_0}$ was available, we can find the most likely tuning knob by searching for the MPE in the space of complete variable instantiations $\{PV, PE, TK\}$ consistent with $\mathbf{e_0}$. By using the simple non-exhaustive search technique discussed in Section IV, this query leads more efficiently to the same result t_1, for which the MPE probability, denoted by MPE_P = $Pr(v_1, p_2, t_1)$=$Pr(v_1) \cdot Pr(t_1) \cdot Pr(p_2|v_1, t_1)$= $0.4 \cdot 0.5 \cdot 0.8$=0.16. This concept will be applied and implemented in a realistic calibration procedure, as detailed in section IV.

IV. Implementation of one-shot self-calibration based on Bayesian Networks

The concept introduced in the previous section will now be extended to a full calibration procedure, analogous to the one presented in fig.1. This section will go in detail through all the necessary steps to achieve this. The complete procedure is depicted in fig.3.

A. Dataset collection

Similar to all SotA methodologies, the first step is to collect the dataset that will be used for the training stage of the model (fig.3(a)). Specifically, the dataset set would be composed of N circuit instances, with variations representative of the overall manufacturing process. For each circuit instance and all K settings of the tuning knobs TK, the process sensor values PV, and circuit performances PE are extracted. Several process sensors and tuning knobs are required to effectively monitor all process variations and enlarge the available tuning space.

B. Discretization

In contrast to typical SotA *regression-oriented* statistical calibration, we rely on a *classification-oriented* approach, in which all variables are discretized. This choice is motivated by its implementation simplicity and consequently hardware efficiency. As illustrated in fig.3(b), this involves the discretization of sensor observations PV, tuning knobs TK and performances PE. Specifically:

Tuning knobs TK**:** they are usually already discrete, each knob taking only a limited number of values.

Process sensors PS**:** Each measurement is split into M bins, according to the probability distribution of the samples. Bin lengths are then non-uniform and follow the sample density over the whole measurement range. This enables to cluster measurements more finely in dense regions (i.e. where we have a lot of samples available).

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:27:21 UTC from IEEE Xplore. Restrictions apply.

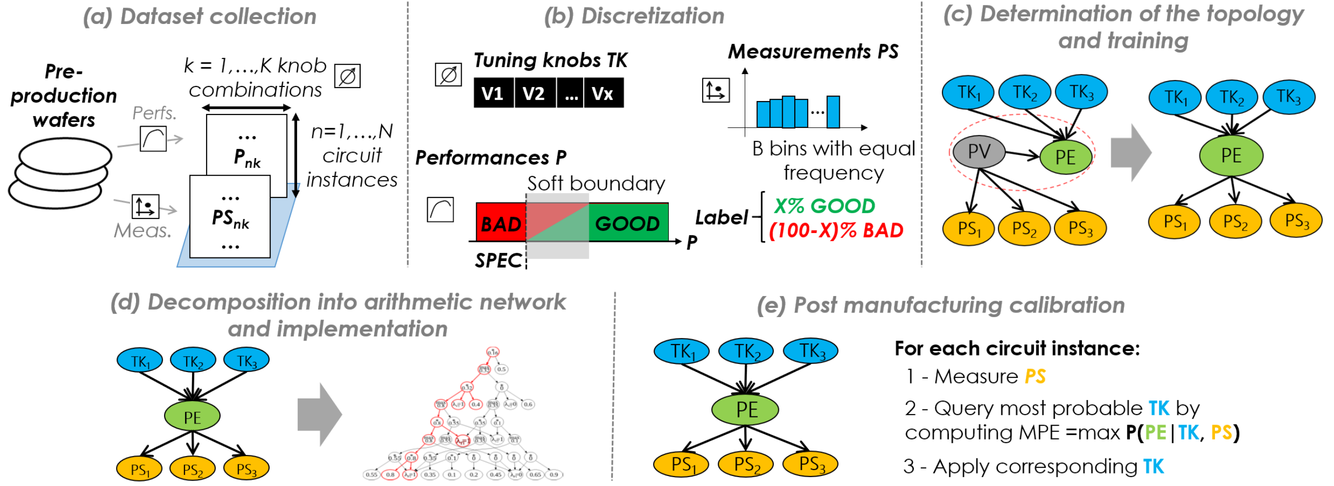

Fig. 3. Illustration of the training procedure

Performances PE: similar to other classification-oriented test techniques, the performance space is divided into two regions. For each target performance variable, two classes exist, representing whether the particular performance satisfies the specifications (the class GOOD) or not (the class BAD). In addition, for this application, it is very valuable to express the level of confidence with which the performance samples are included within the two regions. We utilize the concept of *soft evidence* to build this soft boundary between classes, by recognizing that there is a level of uncertainty to each of the performance samples. Specifically, we label each sample as having $X\%$ of chance to be in the GOOD class and $(100-X)\%$ chance to be in the BAD class, $0 \leq X \leq 100$ (fig.3(b)). As it will be shown in Section V, this enables us to target a specific performance area according to the calibration objectives (more reliability, less power, etc). During BN training, this soft boundary is expressed as a percentage for each sample of the dataset to be labeled in each class.

C. Determination of the topology and training

In the current application context, one of the advantages of BNs over other machine learning paradigms is twofold: (a) BNs easily allow to include expert knowledge, used to determine the network topology, i.e. the nodes and the link between them (although it can also be learned from the data) and (b) BN are particularly efficient to deal with limited data [13], compared to NNs for instance.

a) Determining the topology: we extend the simple example of fig.2 to express the probabilistic relations among the variables available for the current application. In particular, let us consider a circuit with three tuning knobs $TK = \{TK1, TK2, TK3\}$, and three process sensors $PS = \{PS1, PS2, PS3\}$, measured to estimate process variations PV. The performance to optimize, PE, is affected by any change of the tuning knob settings TK, as well as the process variations PV. These dependencies lead to the selected BN topology as depicted in fig.3(c). It is important to note that, similar to regression-oriented approaches [5], process variations PV can not be explicitly extracted and therefore remained un-observed in the proposed BN model (i.e. PV would be a hidden node in the BN). Under these conditions, a complete variable instantiation is not possible anymore, rendering a MPE search infeasible. To circumvent this issue, we exploit the fact that PV changes directly impact PE's behavior and that although unobserved, PV are known through PS. Thus, this assumption enables to merge PV and PE in the model in a single PE node, attaining the final topology a depicted in fig.3(c). We then perform an indirect measurement of PV through PS, which still allows us to know the current conditions and search for the TK configuration that will have the highest likelihood to accomplish the desired PE behavior. It can be noted that tools like [14] can also determine the most suitable BN topology automatically based on the dataset.

b) Training: using this model topology and the dataset created according to Section IV.A, we train the BN model. This consists of estimating the CPT's parameter via a Maximum Likelihood estimation with the generated dataset. For the experiments in this paper, we used Matlab's Bayesian Network toolbox to go through this process [14].

D. Implementation of Bayesian networks using arithmetic circuits

Once the model topology is set and the model is trained, it can be deployed on-chip to perform the actual self-calibration. As mentioned earlier, this involves a MPE search to derive the most probable TK settings, given observations PS and target performances PE. As this process should run on-line and on-chip, the resource efficiency of the MPE search is of crucial importance.

A hardware efficient approach to represent the BN and implement the MPE solution efficiently, is to use Arithmetic Circuits (ACs). An AC is a tractable probabilistic representation that allows to reduce the inference problem to a Weighted Model Counting (WMC) [15]. Such a representation consists of elementary arithmetic operators (sum and product) and maximally exploits the local structural opportunities of the BN

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:27:21 UTC from IEEE Xplore. Restrictions apply.

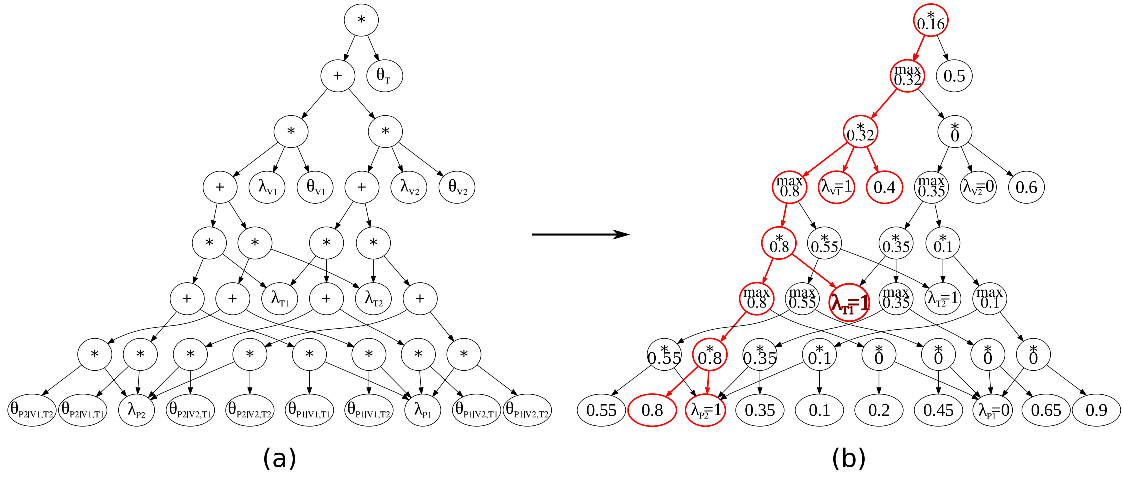

Fig. 4. (a) Arithmetic Circuit. (b) Maximize circuit with MPE sub-circuit.

(e.g. by merging repeated parameters and pruning parameters equal to 1 or 0). The general process is illustrated in fig.3(d). The next paragraph will go into the details of this procedure. Consider again the introductory example depicted in fig. 2. Through the knowledge compilation procedure provided by the ACE package developed by UCLA [16], we can compile the BN into an AC, a graph of add-product operations as depicted in fig. 4 (a). The inputs to this AC are the CPT parameters, represented by $\theta_{x|u}$ and the binary indicator values λ_x, which are set to 1 or 0 depending on whether the value x is observed. Recall that each variable in our example BN can take on two different values (see the node definitions in fig. 2). For our example, the AC variables are thus defined as follows:

- PV: λ_{v_1}, λ_{v_2}, θ_{v_1}, θ_{v_2}
- TK: λ_{t_1}, λ_{t_2}, θ_{t_1}, θ_{t_2}
- PE: λ_{p_1}, λ_{p_2}, $\theta_{P_1|t_1,v_1}$,..., $\theta_{p_1|t_1,v_2}$,..., $\theta_{p_2|t_2,v_2}$

Consider again the observation of evidence $\mathbf{e}_0$ from the example in section III. To incorporate this evidence, we set the following indicator variable values: λ_{p_1}=0, λ_{p_2}=1, λ_{v_1}=1, λ_{v_2}=0 and λ_{t_1}=1, λ_{t_2}=1 (since we do not observe and hence do not know the value that TK will take on). To find the MPE for TK under evidence, we modify the AC into a *maximizer circuit* by replacing addition nodes with maximization nodes, as detailed in [17] and depicted in fig. 4(b). If we now evaluate this maximizer circuit from bottom to top, we see that the value for the root node equals 0.16, which already reveals the value of MPE_P. The TK setting towards this solution can be recovered from a subsequent downward pass through the MPE sub-circuit. In this pass, we go from top to bottom, and include a.) all children when encountering a multiplication node, and b.) the child with the largest value when encountering a maximization node. The result is highlighted in red in fig. 4(b). Finally, the indicator variables in the obtained sub-circuit, reveal the TK settings for the MPE, in this case corresponding to $TK = t_1$. This matches the result inferred in the introductory example.

By using the AC together with the MPE query, it is possible to obtain the most probable tuning knob candidate in a single tuning step. Note that the number of iterations required for this process is not a function of the number of tuning knobs, nor the number of values that each knob can take on, which is in sharp contrast with SotA techniques that use exhaustive searches or an optimizer. In this work, we make a rough estimation of the hardware-cost of executing this MPE search on-chip, by counting the number of required elementary operations. The procedure entails an "upward" evaluation of the maximizer AC and a "downward" search for the MPE sub-circuit. For the current example, the upward step requires 17 multiplications, 7 comparisons (for the max nodes) and 11 parameter fetches from memory. Since the indicator variables are binary, we consider their cost negligible. The downward step takes place in 8 steps, only 3 of which are comparisons. We use this rationale for the hardware-cost estimation in section V-C.

E. Post-manufacturing calibration

Once converted into an AC and maximizer circuit, the BN of fig.3(c) is ready to be used in a post-manufacturing calibration step and be implemented on-chip as shown in fig.3(e). First, the current process sensor values PS are measured by using, for instance, a BIST circuitry. Then, the value of PE is fixed to target a given area, for example to obtain the highest PE possible. Then the MPE search is performed, to find the most probable tuning knob combination considering the evidence on PS and PE. This tuning knob setting TK_{OPT} is then applied to the circuit and the calibration process ends.

Additional benefits of the approach

The proposed technique can be used to include several performance targets. The corresponding nodes and their dependencies with the current variables can be added to the BN. This enables joint tuning knob queries, simultaneously satisfying performance targets along several parameters. This will be demonstrated in section V. Moreover, other measurements or variables can also be included in the BN, such as a measure of aging, temperature, etc. to embed more functionalities on-line.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:27:21 UTC from IEEE Xplore. Restrictions apply.

Limitations of the approach

1) Discretization of variables: This process has to be put in perspective with the computational effort necessary to compute the MPE with the BN, directly reflected in the hardware cost for on-chip implementation. A finer or coarser discretization then results in a trade-off between the BN performance to predict the right tuning knob settings, and the associated hardware cost.

2) Classification instead of regression: In the literature, the development of regression-oriented methodologies over classification-oriented ones has been motivated by their ability to predict directly the performance values, instead of giving a pass/fail decision. In this work, we tend to mitigate this issue by dividing the performance into more than just pass/fail regions, with soft boundaries driven by the trade-off between reliability and performance, as illustrated in Section V. In addition, it should be also noted that in every case that the performances are calibrated without being measured explicitly, a final test may be performed after calibration to confirm whether calibration has succeeded.

V. Case study: self-calibrated low-noise amplifier

To demonstrate the potential advantages of an approach using BNs for self-calibration purposes, an experimental platform containing a RF Low-Noise Amplifier (LNA) is used as a case study. This platform has initially been presented in [3] to compare several one-shot self-calibration methodologies, and used to demonstrate another calibration technique [8] based on-chip analog neural network [18]. The objective is to obtain a fair comparison between existing techniques and the proposed approach, using the same silicon data.

A. Experimental platform and methodology

The experimental platform has been fabricated with IBM's 130 nm CMOS RF process and illustrated in fig.5. The circuit under calibration is a tunable LNA operating at 1.575GHz. Process variations are monitored through different sets of process sensors PS [3]. We use non-intrusive sensors in our case study, as proposed initially in [5]. They consist of one resistor, one capacitor and one NMOS transistor, all copied from the initial circuit topology and placed as test coupons next to the LNA. Since sensors and circuit are in close physical proximity, they witness very similar process variations. The sensor values are as such correlated to the performances of the circuit, even though these sensors are not physically connected to it. Used sensor measurements are simply the value of the resistor, the value of the capacitor and the NMOS transconductance. The LNA is equipped with three voltage bias tuning knobs TK, each varying from 0.8V to 1.4V with a 0.1V interval. As such, each tuning knob can take 7 values, and each LNA can hence be tuned with $K = 7^3 = 343$ knob combinations. The fabricated IC is housed in a custom-designed evaluation board containing four LNAs per die, interfaced with power supplies and external measurement instrumentation. In total, 144 LNA instances have been fabricated, from 36 dies. This enables to obtain a dataset that is statistically representative of the fabrication process.

B. Achieving a "one-shot" tuning process

The proposed one-shot tuning strategy is first compared with the methodology presented in [8].

a) Calibration objective: The general principle is to tune the LNA based on a figure of merit (FoM), representative of a trade off between LNA performances and power consumption. Specifically, the FoM is defined as:

$$FoM = \frac{S_{21}}{(NF-1).P_{DC}} \tag{3}$$

with S_{21} representing the gain, NF the noise figure and P_{DC} the power consumption. More details about the FoM are given in [8]. The best tuning knob setting is referred here as the one that achieves *the highest FoM*.

b) Baseline methodology for comparison: the state-of-the-art reference for this calibration objective [8] trains an on-chip analog NN as a regression to predict the FoM from the process sensor values and the tuning knob settings. The calibration procedure consists on measuring the process sensors on-chip, and then exhaustively going over all possible tuning knob combinations. For each combination, the NN is evaluated to predict the expected performance and find the tuning combination that achieves the highest FoM. To benchmark with this approach, we implement a digital NN in a computer-aided framework off-chip. As highlighted in [8], this approaches achieve the same results as the analog NN.

c) Proposed BN methodology: the dataset is discretized following the procedure described in section IV. The process sensors are discretized in 10 non-uniform bins. The FoM is labeled as 'non-satisfying' (BAD) or 'satisfying' (GOOD), both classes separated by a soft boundary. The probabilities for a sample to lie in the GOOD or BAD class, respectively $P(GOOD)$ and $P(BAD)$, are defined as:

$$P(GOOD) = \frac{FoM - FoM_{MIN}}{FoM_{MAX} - FoM_{MIN}} \tag{4a}$$

$$P(BAD) = 1 - P(GOOD) \tag{4b}$$

With FoM_{MIN} and FoM_{MAX} the respective minimal and maximal FoM values over all circuit instances in the dataset. In a nutshell, the closer the sample is to FoM_{MAX}, the higher its probability is for the GOOD class. The best tuning knob setting is then found by querying the MPE for **TK** in the BN, achieving the one-shot tuning property. The evidence is given as $\{FoM = GOOD, PS1, PS2, PS3\}$. For a fair assessment, a cross-validation technique is used, in which the machine-learning algorithm is trained with all chips except one, and is validated on the left out chip. This procedure is repeated by iteratively putting out every instance in the dataset until all have been tested.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:27:21 UTC from IEEE Xplore. Restrictions apply.

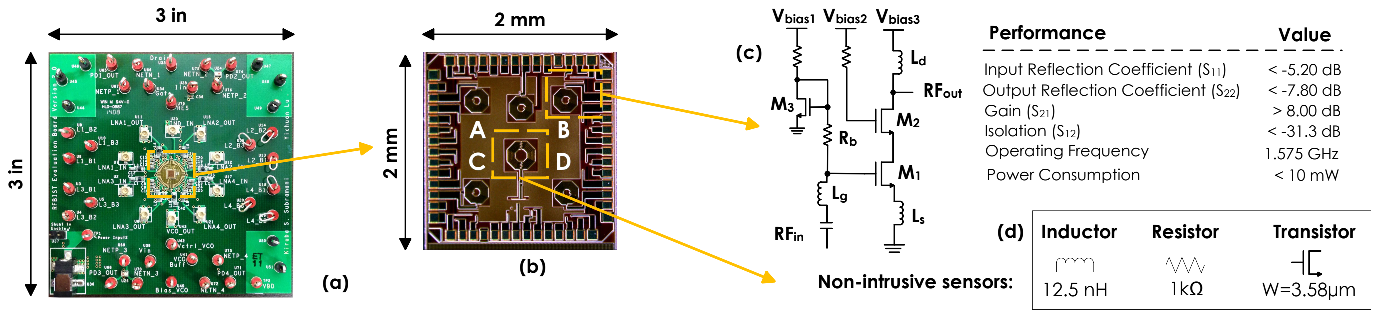

Fig. 5. From [3]: experimental platform (a) evaluation board, (b) microphotograph of the fabricated die, (c) LNA schematic, (b) non-intrusive sensors

d) Results: Fig.6 shows the histogram of performances before and after calibration for both NN and BN approaches, with the overlap between the two methodologies in gray. The initial conditions before calibration are considered with all tuning knobs at their median position (1.1V). It is evident that the two methodologies achieve identical results on the proposed dataset, in terms of performance and power consumption. Their performance is also identical, predicting the correct best FoM for 137 instances out of 144. As an additional benefit, other circuit performances such as $S11$, $S12$ and $S22$, are also improved. While both approaches give the same calibration performance, they will however not come with the same on-chip hardware cost as detailed in the next section.

C. Possibility of efficient hardware implementation

In a second step, the same experiment will be used to compare the hardware cost of both methodologies. For that, we will compare the number of operations that would be required for a digital implementation, for instance in a small processor. This comparison is performed at a high-level and only intends to provide a general overview of both implementation costs.

a) Neural network implementation: To estimate the cost of the NN based approach, we estimate the number of operations of a multi-layer preceptron, as depicted in fig.7, composed of: (a) one input layer, with N_I nodes, (b) one hidden layer with N_H neurons, and (c) one output layer with N_O neurons. Following the strategy in [8], the tuning knob combinations are searched exhaustively to find the best FoM. To obtain one FoM value, the NN requires one multiplication, one addition and one memory fetch per incoming edge in every neuron, and one multiplication, one addition and one memory fetch per neuron (except in the input layer) to implement the activation function. This represents a total of $(N_I - 1) * N_H + (N_H - 1) * N_O + N_H + N_O$ additions and $N_I * N_H + N_H * N_O + N_H + N_O$ multiplications. By using 6 inputs, 10 neurons in the hidden layer and one output, as in [8], the number of operations can be estimated to 74 additions 81 multiplications, and 81 memory fetches. This value has to be multiplied by the number of tuning knob combinations searched for, in this example 343. The total cost of running the complete NN-based search, is given in the column NN_{TOT} of Table I.

b) Compilation of the general BN: as explained throughout Section IV-D we propose to implement the MPE solution with an AC, which consists of 1617 nodes of which 856 are multiplications, 161 additions (or comparisons, after the maximizer circuit has been generated), 53 indicator variables and 547 parameter variables. This defines the cost of the upward pass through the network (cfr. procedure explained in Section IV-E). The downward pass requires significantly less evaluations as only one path through the network is assessed. E.g. when assuming e=$\{PE{=}GOOD, PS_1{=}2, PS_2{=}1, PS_3{=}4\}$, the downward pass requires only 16 steps, 7 of which are comparisons. These results are summarized in the column BN of Table I.

c) Compilation of the BN with evidence: further cost reduction can be obtained by making application specific assumptions. For example, we can compile an AC with evidence [16] (because we are only interested on $PE = GOOD$), and considerably reduce the number of required operations as shown in the column BN_{EV} of Table I.

d) Comparison: To obtain the total aggregated cost of each approach, we consider a relative cost per operation derived for a 32 bit floating point format in a 45nm CMOS process, as derived from [19] and summarized in the last column of Table I. With this information, the total cost of every approach is computed, and summarized in the last line of Table I. It is clear that both the BN based approaches have a significantly lower hardware cost than the NN_{TOT} approach. Clearly, this is a consequence of the exhaustive search needed for the NN based implementation, which is circumvented by finding only the MPE in both the BN based solutions. Effectively, this approach decreases the calibration cost by a factor of 43.5x and 85.7x over the NN. One could argue that a NN can be used with a smarter search procedure than an exhaustive search. E.g. an optimizer could be used to smartly query only selected tuning setting combinations. This would allow to find an optimal solution with less NN queries. Yet, as seen in Table I, a single neural network evaluation is only 7.8x more efficient than the complete BN, or 4x regarding the BN with evidence approach. As such, this approach would only be more efficient if the optimizer can find the optimal setting in less than 4 queries, which is very difficult for the given case study of 3 tuning knobs with 7 settings each. These results confirm that for a similar implementation, the BN-based approach would reduce drastically the number of operations compared to SotA techniques with digital machine-learning implementation.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:27:21 UTC from IEEE Xplore. Restrictions apply.

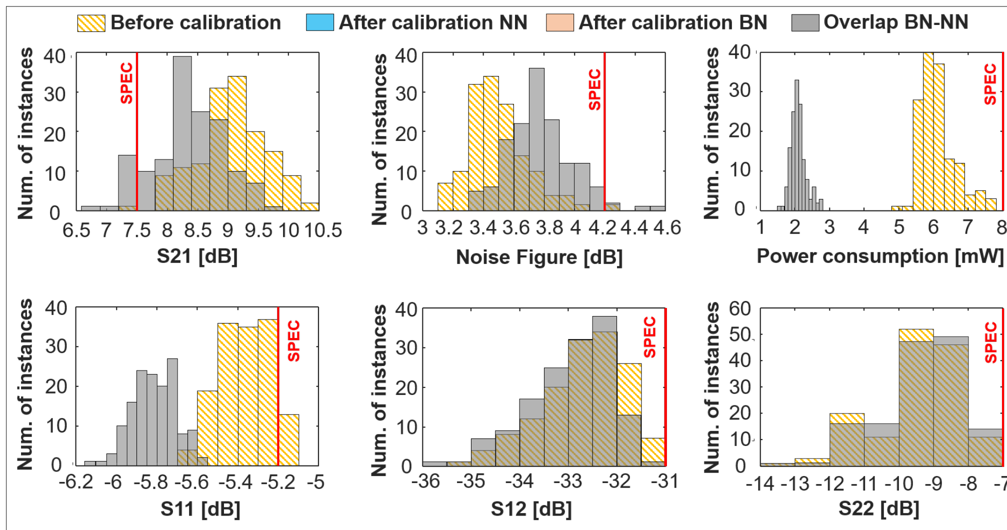

Fig. 6. Comparative of performances after calibration with Bayesian Network (BN) and Neural Network (NN) approaches

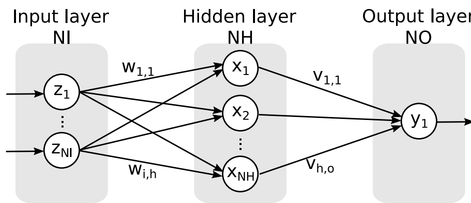

Fig. 7. Neural network used for the cost comparison

TABLE I
HARDWARE-COST ESTIMATION

Operation type	NN	NN_{TOT}	BN	BN_{EV}	Relative op. cost
Add.	74	27440	168	60	1
Mult.	81	29841	856	418	4
Mem.	81	29841	547	296	5
Tot. cost	803	275429	6327	3212	

D. One-shot tuning with several explicit performances

In the previous experiment, the optimization performance target was a FoM, that implicitly includes the three performances (S_{21}, NF and P_{DC}). However, it cannot be ensured that for the best FoM all performances still individually satisfy the specification. This explains why, although this methodology gives very good results in terms of performance enhancement, 17 instances out of 144 still do not satisfy the specifications after calibration, for a yield of 88.2%. This can be circumvented by training three different NNs to explicitly predict S_{21}, NF and P_{DC}. This would however further increase the cost of the technique since optimizer or an additional decision circuitry is also required, as discussed previously. In the proposed BN based methodology, those three performances can be included directly and explicitly in a single BN, as depicted in fig.8(left). In addition, the one-shot tuning property stays valid, since the MPE can be queried in the exact same way as for the previous experiment.

a) Calibration objectives: in this experiment, the objective is now to find the tuning knob setting that maximizes the reliability, i.e. the probability that each performance lies as far as possible from its specification line.

b) Proposed methodology: the methodology is similar to the one described in section V-B, except that in this case S_{21}, NF and P_{DC} are considered explicitly in the BN. This in turns leads to define several soft boundaries, one for each performance, as shown in fig.8(right). They are defined as:

$$P_{S21}(GOOD) = \begin{cases} \frac{S21_S - S21_c}{S21_{MAX}}, & \text{if } S21 > S21_S \\ 0, & \text{otherwise} \end{cases}$$

$$P_{NF}(GOOD) = \begin{cases} \frac{-(NF_S - NF_c)}{NF_{MIN}}, & \text{if } NF < NF_S. \\ 0, & \text{otherwise} \end{cases}$$

$$P_{PDC}(GOOD) = \begin{cases} \frac{-(PDC_S - PDC_c)}{PDC_{MIN}}, & \text{if } PDC < PDC_S \\ 0, & \text{otherwise} \end{cases}$$

with V_{MIN}, V_{MAX}, V_S and V_c representing respectively the minimal value, maximal value, the specification and the current value of variable V. For each variable, $P_V(BAD) = 1 - P_V(GOOD)$. Essentially, the further away from the specification line, the greater the probability to be in the GOOD class. Again, the optimal tuning knob setting is assessed by querying the MPE in one-shot to achieve the combined goal cross all target performance variables S_{21}, NF and P_{DC} lying in the GOOD class, given the observed PS.

c) Results: the histograms of the LNA performances after this calibration are depicted in Fig.9. The proposed methodology is compared with the previous experiment where only the FoM is predicted. As can be observed, the yield loss is significantly reduced since only one circuit out of 144 does not satisfy all specifications, increasing yield up to 99.3%. The price to pay for this increased yield is an increase of the power consumption, as now the calibration is targeted for maximum reliability. However, the power consumption stays significantly lower than the specification.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:27:21 UTC from IEEE Xplore. Restrictions apply.

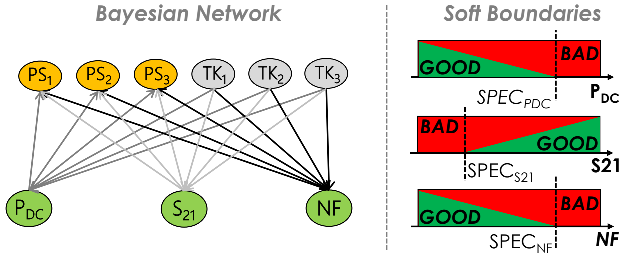

Fig. 8. Bayesian networks and soft boundaries used in section V-D

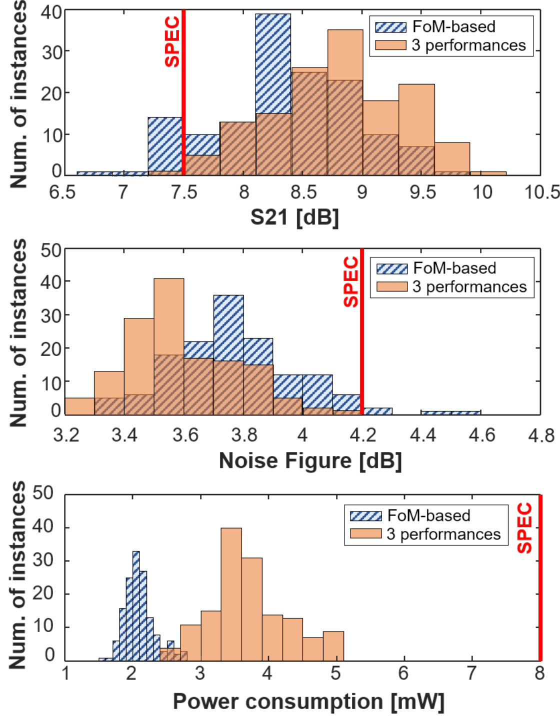

Fig. 9. Comparative of performances after calibration for FOM based calibration and the proposed methodology

d) Hardware cost: the BN presented throughout this section can also be compiled to an AC and transformed to a maximizer circuit to assess the cost of the MPE query. In particular, evaluating the maximizer circuit for this setup requires 3383 multiplications, 479 comparisons and 1231 parameter fetches, while finding the MPE requires 10 comparisons. One again compiling the BN with evidence (for the three performance nodes = GOOD), renders a reduced AC, with 434 multiplication, 60 addition and 608 memory fetch operations for the upward evaluation and 6 comparisons for the MPE search. This cost is hence very close to the cost of the FoM-based query of section V.C. This is in sharp contrast to traditional approaches, whose cost blows up when assessing several performance variables in parallel.

E. Calibrate with different constraints using soft boundaries

As a final experiment, the same problem and network structure will be considered but with several calibration objectives to make a more balanced trade-off between power consumption optimization, and reliability.

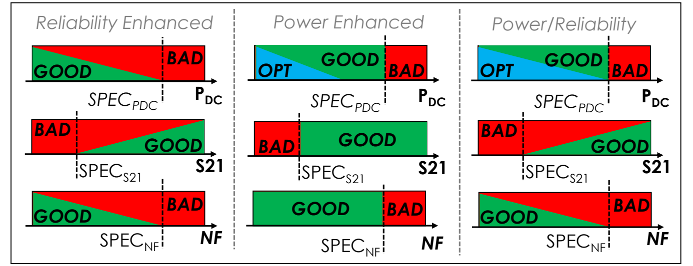

Fig. 10. Illustration of the soft boundaries used for the three calibration objectives

TABLE II
COMPARATIVE OF THE YIELD LOSS AND MEAN POWER CONSUMPTION FOR THREE CALIBRATION OBJECTIVES: RELIABILITY ENHANCED, POWER ENHANCED AND POWER/RELIABILITY TRADE OFF

Calibration	Yield Loss	Mean Power [mW]
Rel. enhanced	1/144	4.22
Pow./Rel. trade-off	7/144	2.43
Pow. enhanced	14/144	2.09

a) Calibration objectives: By using several soft boundaries it is possible to drive the tuning knob search towards several performance regions. Three cases will be considered:

Reliability enhanced: all performances as far as possible from their specification line. This is the same as the previous experiment of section V-D.

Power enhanced: push power consumption as low as possible with limited yield impact.

Power/reliability trade-off: find a compromise between reliability and power constraints.

b) Proposed methodology: the three cases are achieved using variations on the soft-boundary approach. The boundaries used for the three considered case studies are illustrated in fig.10. For the reliability enhanced case, the soft boundaries are the same as in the experiment section V-D. For the power enhanced case we introduce a third class, named as 'OPT', referring to the optimal power consumption target. The aim is to push power down as much as possible. We do this by fixing the evidence in the BN for the power consumption to be OPT during the MPE search. As compensation, we fix the boundaries for $S21$ and NF as hard ones, following the specification line. For the power/reliability case, the power is pushed less towards the minimum by extending the OPT class soft boundary as in fig.10(right), while keeping soft boundaries for the other performances variables pushing them again away from the specification limits.

c) Results: a comparison of obtained performances for these three cases is depicted in fig.11 and in Table II. As it can be observed, the power enhanced approach significantly reduces power consumption, as the FoM-based calibration. The reliability enhanced scenario corrects the yield with an impact on power consumption, as explained in section V-D. A compromise between both approaches is achieved in the power/reliability trade-off, when the yield is slightly impacted, but the mean power is divided by almost a factor 2. The search of the best tuning knob setting can as such be driven in a continuum between high reliability and low-power.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:27:21 UTC from IEEE Xplore. Restrictions apply.

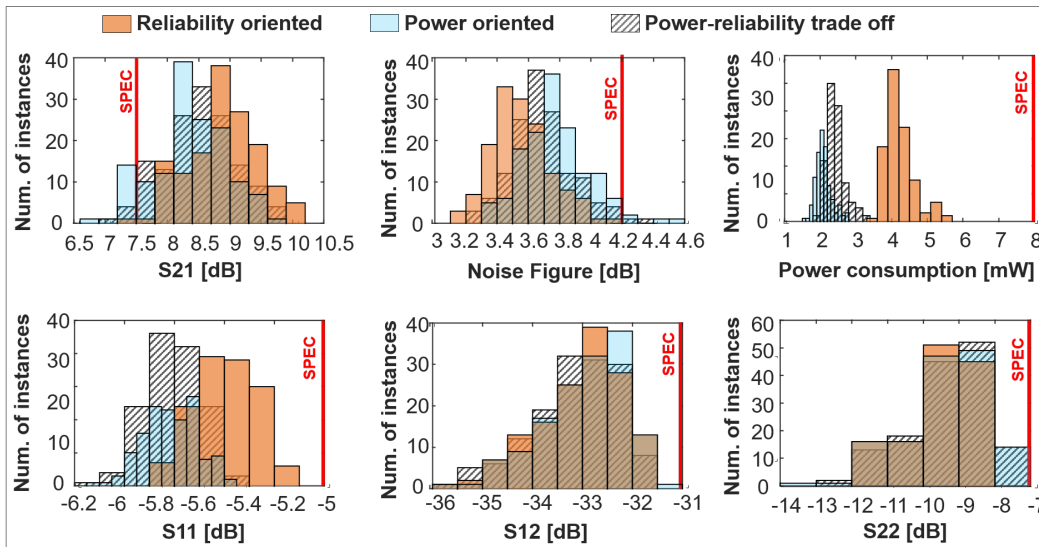

Fig. 11. Comparative of performances after calibration for power enhancement, reliability enhancement and power/reliability trade-off

VI. Conclusion

This work presents a new direction towards efficient statistical one-shot on-chip calibration of analog/RF circuits, promoting the use of Bayesian Networks for statistical self-calibration. The proposed calibration scheme has been validated on an experimental platform containing 144 low-noise amplifiers fabricated in 130nm CMOS technology. The results demonstrate 1.) on par performance compared to other SotA self-calibration techniques using neural networks; 2.) at a lower general cost; 3) with the additional capability to easily incorporate several co-existing performance targets and make balanced trade-offs between performance reliability and power consumption savings.

VII. Acknowledgements

Prof. Verhelst's, Dr. Andraud's and Mrs. Galindez's participation in this research has been supported by the European Union's Horizon 2020 research and innovation program under grant agreement No 665347, and by European Research Council (ERC) grant program in the Re-SENSE project under grant agreement n° 715037.

Prof. Makris' and Mr. Lu's participation in this research has been partially supported by the Semiconductor Research Corporation under contract SRC 2625.001 and the National Science Foundation under contract NSF 1527460.

References

[1] F. Poehl, F. Demmerle, J. Alt, and H. Obermeir, "Production test challenges for highly integrated mobile phone SOCs: A case study," in *2010 15th IEEE European Test Symposium*, May 2010, pp. 17–22.

[2] P. N. Variyam, S. Cherubal, and A. Chatterjee, "Prediction of analog performance parameters using fast transient testing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 3, pp. 349–361, 2002.

[3] Y. Lu, K. S. Subramani, H. Huang, N. Kupp, and Y. Makris, "A comparative study of one-shot statistical calibration methods for analog/RF ICs," 2015, pp. 1–10.

[4] D. Banerjee et al., "Self-learning RF receiver systems: Process aware real-time adaptation to channel conditions for low power operation," *IEEE Transactions on Circuits and Systems-I: Regular Papers*, vol. 64, no. 1, pp. 195–208, 2017.

[5] M. Andraud, H.-G. Stratigopoulos, and E. Simeu, "Post manufacturing one-shot calibration of RF circuits based on non-intrusive sensors," *IEEE Transactions on Circuits and Systems-I: Regular Papers*, 2016.

[6] N. Kupp, H. Huang, Y. Makris, and P. Drineas, "Improving analog and RF device yield through performance calibration," *IEEE Design & Test of Computers*, vol. 28, no. 3, pp. 64–75, 2011.

[7] D. Han, B. S. Kim, and A. Chatterjee, "DSP-Driven self-tuning of RF circuits for process-induced performance variability," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 2, pp. 301–314, 2010.

[8] G. Volanis, D. Maliuk, Y. Lu, K. S. Subramani, and Y. Makris, "On-die learning-based self-calibration of analog/RF ICs," 2016.

[9] S. Lee, C. Shi, J. Wang, S. Sanabria, H. Osman, J. Hu, and E. Sanchez-Sinencio, "A built-in self-test and in situ analog circuit optimization platform," *IEEE Transactions on Circuits and Systems-I: Regular Papers*, pp. 1–14, 2018.

[10] E. J. Wyers, M. A. Morton, T. C. L. C. Sollner, C. T. Kelley, and P. Franzon, "A generally applicable calibration algorithm for digitally reconfigurable self-healing RFICs," *IEEE Transactions on Very Large-Scale Integration Systems*, vol. 24, no. 3, pp. 1–14, 2018.

[11] T. Das, A. Gopalan, C. Washburn, and P. R. Mukund, "Self-calibration of input-match in RF front-end circuitry," *IEEE Transactions on Circuits and Systems-II: Express Briefs*, vol. 52, no. 12, pp. 821–825, 2005.

[12] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[13] D. Georges et al., "A generative vision model that trains with high data efficiency and breaks text-based CAPTCHAs," *Science*, vol. 358, no. 6368, 2017.

[14] K. P. Murphy, "The bayes net toolbox for MATLAB," *Computing Science and Statistics*, vol. 33, p. 2001, 2001.

[15] M. Chavira and A. Darwiche, "On probabilistic inference by weighted model counting," *Artificial Intelligence*, vol. 172, no. 6-7, pp. 772–799, 2008.

[16] UCLA. Ace compiler. [Online]. Available: http://reasoning.cs.ucla.edu/ace/

[17] H. Chan and A. Darwiche, "On the robustness of most probable explanations," *arXiv preprint arXiv:1206.6819*, 2012.

[18] D. Maliuk and Y. Makris, "An experimentation platform for on-chip integration of analog neural networks: A pathway to trusted and robust analog/RF ICs," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 8, pp. 1721–1734, 2015.

[19] M. Horowitz. Energy table for 45nm process, stanford VLSI wiki. [Online]. Available: https://sites.google.com/site/seecproject

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:27:21 UTC from IEEE Xplore. Restrictions apply.