2022 18th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD) | 978-1-6654-6703-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/SMACD55068.2022.9816185

Mixed-Variable Bayesian Optimization for Analog Circuit Sizing using Variational Autoencoders

Konstantinos Touloupas and Paul P. Sotiriadis
National Technical University of Athens, Greece
E-mail: ktouloupas@mail.ntua.gr

***Abstract*—Bayesian Optimization (BO) has recently gained popularity within the context of automatic sizing of analog and Radio-Frequency (RF) Integrated Circuits (ICs). However, its reliance on Gaussian Process models, which operate only on continuous-valued spaces, reduces its applicability in real-world scenarios, where multiple discrete-valued variables often exist. In this paper, we propose an approach to mitigate this issue by using a Deep Learning scheme to transform devices parametrizations to continuous ones, where classic BO can be applied. Specifically, a composite architecture that consists of a Convolutional Variational Autoencoder (VAE) and a dense Neural Network is built to define a continuous representation of integrated inductors in a TSMC 90nm process. By optimizing using these representations, we overcome the limitation of discrete-valued variables. Experimental results on a Low Noise Amplifier highlight the efficiency of the proposed approach.**

***Index Terms*—sizing, optimization, analog, deep learning**

I. Introduction

Methods for automatic sizing of analog and Radio Frequency (RF) circuits constitute an important challenge for the semiconductor industry. Due to the continuous integration of Integrated Circuits (ICs), designers have to extensively verify their circuits while accounting for short-channel effects as well as process, voltage and temperature (PVT) variations. This renders the design cycle time consuming or leads to inefficient and error prone designs [1]. Therefore, the development of EDA tools that automate the design of complex systems has merit and is an active research topic [1]–[3].

In the context of automatic analog circuit design, the state of the art approaches are simulation-based optimization ones [1], [4]–[6]. They formulate the circuit sizing problem as an optimization one, where parametrized testbenches are simulated using commercial simulators, to yield the device sizes that satisfy user-provided constraints. By relying on simulators and foundry-provided models, these approaches typically yield accurate results. The optimization is carried out by a black box optimization algorithm, such as Evolutionary Algorithms (EAs) [1]. While EAs have been successfully applied for analog circuit sizing, they typically require a large number of evaluations, which may be time-consuming in the case of analog circuit simulations. Bayesian Optimization (BO) [7] has recently gained popularity as an alternative, since it typically requires fewer evaluations to reach optimal solutions, with some of its variants successfully applied to automated analog circuit sizing [4]–[6].

BO is an iterative surrogate modelling procedure that uses Gaussian Processes (GPs) [8]. GPs, however, operate only on continuous spaces [8] and subsequently hinder the application of BO in problems that include discrete variables. In practice, many real-world circuits include devices such as integrated inductors, which are parametrized by both discrete and continuous variables.

Motivated by the above, we present an approach to parameterize IC devices using continuous representations and demonstrate it on integrated inductors. We build a deep learning architecture that is comprised of a Convolutional Variational Autoencoder (VAE) [9] and a fully connected Neural Network (FCNN). By training the VAE on the values of Quality Factor and Inductance over a set of frequencies, a data-driven representation of inductors is defined on a continuous space. The FCNN is trained to map the continuous representation back to the geometric sizes of the inductors. Given the above architecture, we formulate the automatic sizing of circuits using the continuous representations and solve it using BO. As a proof of concept, a Low Noise Amplifier (LNA) is automatically sized using the continuous inductor representations.

The paper is organized as follows. In Section II, the employed BO algorithm is discussed. The proposed Deep Learning architecture for the continuous parametrization of inductors is presented in Section III. Finally, Section IV presents an application example on a variation-aware BO optimization of an LNA and Section V concludes the paper.

II. Employed Bayesian Optimization

Consider the case of a Single-Objective (SO) optimization problem with constraints

$$\begin{aligned} \min \quad & f(\mathbf{x}), \quad \mathbf{x} = [x_1, x_2, \ldots, x_d] \\ \text{s.t.} \quad & g_j(\mathbf{x}) \leq 0, \quad j = 1, \ldots, l, \quad \mathbf{x} \in \mathbb{S} \end{aligned} \tag{1}$$

where the objective function $f(\cdot)$ and the constraint functions $[g_j(\cdot)]_{j=1}^{l}$ are time-consuming to evaluate. BO is a sample-efficient approach to solve the above problem, by using surrogate models to balance the exploration-exploitation tradeoff of the loss landscape [7].

BO consists of two main components; the GP surrogate models, which approximate the objective and constraint functions of Eq. (1), and an acquisition function $a : \mathbb{S} \to \mathbb{R}$. This uses the GPs' probabilistic information to assign a score of utility to each point in $\mathbb{S}$, i.e. it predicts the goodness of

978-1-6654-6703-2/22/$31.00 ©2022 IEEE

each point in the variable space. BO works in iterations by repeating the steps of function(s) evaluation, GP model update and acquisition function optimization. By maximizing a, the maximum utility point is derived and it is chosen as the query point for evaluation in the next iteration.

The GP models employed within BO are stochastic processes of infinitely many random variables and are defined by their mean function $m : \mathbb{S} \to \mathbb{R}$ and their kernel function $k : \mathbb{S} \times \mathbb{S} \to \mathbb{R}$ [8]. Any finite set of these random variables is jointly Gaussian. After training on a set on input-outputs $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$ of the function to-be-minimized, they provide a L-dimensional Gaussian posterior distribution over $L \geq 1$ vectors $\{\mathbf{x}_i^\star \in \mathbb{S}\}_{i=1}^{L}$ with mean $\boldsymbol{\mu} = [\mu_i]_{i=1}^{L}$ and a $L \times L$ covariance matrix $[\mathrm{Cov}(i,j)]_{i,j=1}^{L}$ such that [8]

$$\begin{aligned} \mu_i &= \mathbf{k}_i^{\mathrm{T}} K^{-1} \mathbf{y} \\ \mathrm{Cov}(i,j) &= k(\mathbf{x}_i^\star, \mathbf{x}_j^\star) - \mathbf{k}_i^{\mathrm{T}} K^{-1} \mathbf{k}_j. \end{aligned} \tag{2}$$

Here, $\mathbf{k}_i^{\mathrm{T}}$ is a $(1 \times N)$ vector $[k(\mathbf{x}_j, \mathbf{x}_i^\star)]_{j=1}^{N}$ and $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) + \sigma_n^2 \delta_{ij}$, where σ_n^2 is a GP hyperparameter [8] and δ the Kronecker delta. This equation is utilized by a to define the utility of each point in $\mathbb{S}$.

While vanilla BO acquisition functions assume $L = 1$ in Eq. (2) and provide with a single query point at each iteration, recent works [5], [6] proposed techniques to acquire a batch of query points, in order to exploit any infrastructure that can run multiple simulations at the same time. In [5], a Thompson sampling [7] method was used to build analytic approximations to function samples from the distribution in Eq. (1), in the case of multi-objective constrained optimization. Here, the aforementioned method is adapted for SO optimization with constraints; For each of the GP models approximating the functions in Eq. (1), $N_S \geq 1$ analytical samples are drawn. These are fast-to-evaluate [5] and are used in N_S separate SO optimization problems solved by a Genetic Algorithm (GA) [10], which is coupled with the feasibility rule to account for constraints [10]. When all of the auxiliary optimizations are finished, N_S query points are derived and fed to the simulator for parallel evaluation. The reader is referred to [5], [11], for more information on the Thompson sampling procedure.

III. Proposed Deep Learning Architecture

In this section the proposed architecture for inductor continuous representation is discussed and implementation details are provided.

A. Variational Autoencoders

A VAE [9] is a graphical model that, when trained on a real-world dataset $\{\mathbf{x}_i \in \mathbb{X}\}_{i=1}^{N}$, can produce synthetic samples that follow approximately the distribution of the inputs $\{\mathbf{x}_i\}_{i=1}^{N}$. In probabilistic terms, the data generation is a two-step process wherein a continuous latent variable $\mathbf{z} \in \mathbb{Z}$, that follows a prior distribution $p(\mathbf{z})$, is sampled, and the synthetic data are generated by sampling from the conditional distribution $p(\mathbf{x}|\mathbf{z})$. In the VAE case, the dataset inputs $\mathbf{x}$ are mapped to values $\mathbf{z}$, which are the latent variables used to generate data using the aforementioned procedure.

A VAE consists of two parts; an probabilistic *encoder* which maps the inputs to $\mathbb{Z}$ and learns an approximate posterior distribution $q_\theta(\mathbf{z}|\mathbf{x})$ and a probabilistic *decoder* which takes the latent variable values as inputs and learns an approximate likelihood $p_\phi(\mathbf{x}|\mathbf{z})$. The decoder is the generative part of VAE that produces the synthetic samples once fed with a latent variable. The encoder enforces a conditional distribution on the latent variables, given the inputs $\mathbf{x}$. Both of the encoder and decoder are implemented as neural networks, with their weights and biases denoted as θ, ϕ respectively.

Training a VAE to produce both good reconstructions and good approximation of the true density $p(\mathbf{x})$ requires the maximization of the Evidence Lower Bound [9] defined as

$$\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\theta(\mathbf{z}|\mathbf{x})}\Big[\log p_\phi(\mathbf{x}|\mathbf{z})\Big] - \mathrm{D}_{KL}\Big(q_\theta(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})\Big), \tag{3}$$

where the regularization term $\mathrm{D}_{KL}(\cdot,\cdot)$ is the KL-divergence [9] and penalizes the the distance between the approximate posterior $p_\theta(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z})$. The first term is a reconstruction loss that penalizes decoder outputs that do not resemble the inputs to the encoder. In order to take advantage of an analytical derivation for the KL-divergence term, $p(\mathbf{z})$ is set as a zero-mean and unit-variance isotropic Gaussian distribution, whereas $p(\mathbf{z}|\mathbf{x})$ is parametrized as an isotropic Gaussian [9].

B. Latent Space Optimization

A recently proposed method to handle high-dimensional optimization problems involves substituting the original variable space with the latent space of a deep generative model [12], [13]. This requires training a generative model that maps latent space vectors back to original input space, such as the VAE. In the case of a mixed-variable input variable space, this technique maps input-vectors to a continuous representation in the latent space $\mathbb{Z}$. Therefore, an optimization algorithm such as BO can be used to search over $\mathbb{Z}$ for optimal variables, which are then reconstructed by the decoder network and eventually fed to the simulator for evaluation.

In this setting, the structure of the latent is space is important for the formulation of the optimization problem and to attain good results. Taking into account that $p(\mathbf{z})$ is a zero-mean, unit variance isotropic Gaussian, $p_\theta(\mathbf{z}|\mathbf{x})$ shall also resemble to this and therefore the vast majority of mapped input vertices (99.7%) will reside in the $[-3,3]^d$ region, where d is the latent space dimension, die to the Gaussian's properties. This serves as bounds for the latent variable space. Another benefit of VAEs in the latent space optimization scheme is that their loss function induces an ordering of the latent space, such that input vectors that are close in the data manifold are mapped to close locations in the latent space, whereas dissimilar ones are spread out [12]. The latter is important for GPs, which use euclidean distance in the latent space to measure the correlations of potential query points.

C. Architecture

In the case of integrated inductors, we wish to build a continuous representation that both maps latent space vectors

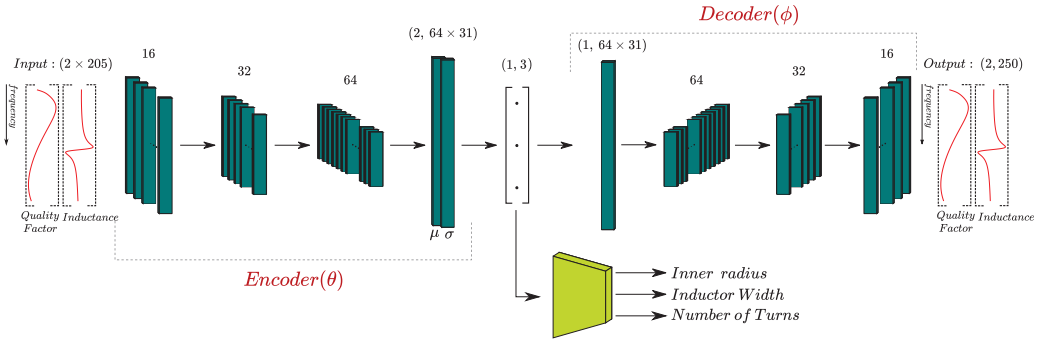

Fig. 1: A depiction of the proposed architecture. 1D vectors of inductors' Quality Factor and Inductance frequency behavior are inputs to the 1D convolutional filters of the architecture. The filter sizes are shown as well. The predictor FCNN gets as input the latent representation of the inductor's frequency characteristics and yields its geometric sizes.

to inductor geometric variables and, ensures that inductors with similar frequency behaviors are mapped to nearby vectors in the latent space. The latter is important to ensure accurate results of the optimization algorithm. To this end we proposed a composite architecture shown in Fig. 1. We consider the case of spiral inductors in a TSMC 90nm process and execute parametric sp simulations in the frequency range $[0.1, 100]$GHz to obtain the s-parameters of 6000 inductor shapes, which are then transformed to inductances and quality factors by [14]

$$L(f) = \frac{\text{imag}(Z_{11})}{2\pi \cdot f}, \quad Q(f) = \frac{\text{imag}(Z_{11})}{\text{real}(Z_{11})}, \tag{4}$$

where Z_{11} is computed from the s-parameters of each simulation [14]. These data are processed to have length 250 and are fed as inputs to a VAE model. This learns to map frequency vectors to a 3D latent space and ensures that inductors with similar $L(f)$, $Q(f)$ are mapped nearby. Note that the same would not apply in the case where inductor geometric sizes were used as inputs to the VAE, since for instance adding another turn in a spiral inductor may significantly affect its behavior. To retrieve the geometric sizes of inductors (inner radius, inductor width, number of turns) from the continuous space, we train a FCNN as a predictor jointly with the VAE.

Since the VAE's inputs are 1D vectors of frequency behaviors, we adopt a convolutional architecture to benefit from the spatial relations between different data points. Note that this would not hold if a fully connected architecture was utilized instead. The encoder network consists of 3 1D convolutional layers with 16, 32, and 64 filters each, while all of them have kernel size 4, stride 2 and padding 1. The ReLU activation function was used. As shown in Fig. 1, the mean μ and variance σ^2 of the Gaussian $p(\mathbf{z}|\mathbf{x})$ are produced by two linear layers as shown, while the decoder has a similar structure as the encoder. The loss function for the VAE part is

$$L_{VAE} = \| \mathbf{x} - \hat{\mathbf{x}} \|_2 + \mathrm{D}_{KL}\Big[\mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x})) \parallel \mathcal{N}(0, \mathcal{I})\Big], \tag{5}$$

where $\mathcal{I}$ the identity matrix and $\hat{\mathbf{x}}$ is the decoder's output. Details on the KL-divergence derivation can be found in [9].

The FCNN, shown in yellow in Fig. 1, has three layers with 3, 50 and 25 neurons each and uses ReLU activations. To predict the number of turns out of 20 possible options in the range $[0.5, 0.75, 1, \ldots, 5.25]$, we use a classification formulation and define a cross entropy loss L_{NoT} to train the associated weights. A separate cross entropy loss L_{IW} applies for the four separate inductor widths that the employed technology provides for spiral inductors, $\{3, 6, 9, 15\}\mu$m and correspond to 4 outputs of the FCNN. Finally, the inductor's inner radius is a continuous variable in the range $[15, 90]\mu$m and its loss is a Mean Square Error (MSE) one, denoted by L_{IR}. The overall loss function for the composite network is:

$$L_{TOTAL} = L_{VAE} + L_{NoT} + L_{IW} + L_{IR}. \tag{6}$$

The model was trained using the Adam optimizer for 1000 epochs and a 80%-20% training-test split. After training, the test data were mapped to their latent representations and predictor's accuracy for number of turns and width is 94% and 96% respectively. For the inner radius, the MSE score is 0.11, where the values are normalized in the $[0, 1]$ range. Fig 2 depicts a reconstructed inductance curve, along with the predicted geometry for a real spiral inductor using the proposed scheme.

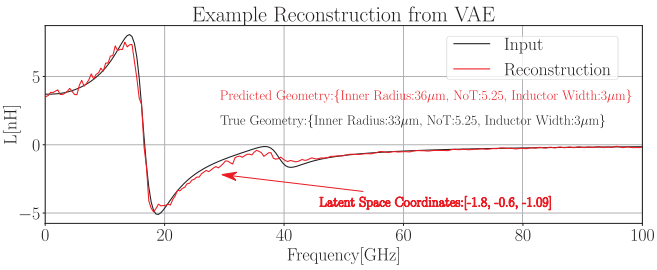

Fig. 2: A reconstruction of inductor's inductance across the frequency range of interest.

IV. Example Optimization

To test the proposed Deep Learning scheme in an optimization setting, we consider the LNA shown in Fig. 3. There are three inductors which are parametrized by the latent variables described in the previous section, while transistors are paremetrized by their lengths and widths, capacitors C_g,

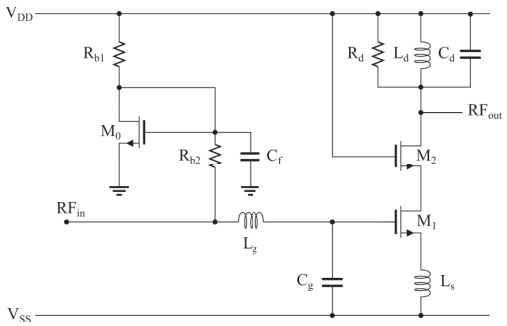

Fig. 3: The LNA considered as a case study.

C_d are crtmom devices parametrized by their fingers widths and spacings and the rppolywo resistor R_{b2} is parametrized by its width and length. The supply is 1.2V and the operating frequency 2.4GHz. The rest of the sizes are constants and in total there are 17 variables.

We wish to minimize static power consumption P_{dc} in the nominal operating conditions, while enforcing $IP3 \geq -5$dBm, $NF \leq 2.5$dB, $S_{11} \leq -8$dB, $S_{22} \leq -8$dB at the operating frequency at the following corners: ss, sf, fs, and ff and at the working temperatures of -50, 27 and 125 Celcius. Also for the nominal conditions we enforce $S_{21} \geq 21$dB. This amounts to a total of 13 testbenches. For comparison we consider the BO discussed in Section II with N_S=5 and 1000 total evaluations, operating on the latent space as well as a variant that relaxes discrete variables to continuous, as used in [5]. In addition, we use a GA with mixed real and discrete crossover and mutation operators, 100 population and 50 generations. The experiments were repeated 5 times to account for random fluctuations.

The results of the optimizations are given in Table I, in terms of the objective P_{dc} and the constraints, where the worst attained performances out of all corners and successful repetitions are depicted. There are two takeaways from this table; first, the BO using the proposed parametrization for inductors is successful and produces better results than the simple rounding method of the relaxation transform BO. Second, both of the BO approaches yield better results compared to the population-based GA in the provided simulation budget, which finds feasible solution only 2 out of 5 times. Therefore, it makes sense to for BO in cases were one cannot afford many circuit simulation due to time restrictions.

V. Conclusion

A Deep Learning Framework was proposed to parametrize integrated spiral inductors with continuous variables, using a convolutional VAE and a FCNN as predictor for inductor sizes. By mapping inductor frequency responses in the VAE's latent space and training the FCNN to reproduce its geometric sizes, we were able to render the sizing problem of a LNA a continuous one and benefit from BO's efficiency to solve a variation-aware sizing problem. The parametrization scheme was shown to enhance the search capabilities of BO, compared to other approaches.

TABLE I: LNA Optimization Results: Mean $\pm$ STD

Method	P_{dc} [mW]	S_{21} [dB]	S_{11} [dB]	S_{22} [dB]	$IP3$ [dBm]	NF [dB]	Success
BO	**11.24 $\pm$ 1.1**	21.35	-8.32	-9	-2.9	2.45	**5/5**
BO-Relaxation	14.36 $\pm$ 2	21.12	-8.4	-9.2	-3.4	2.44	5/5
GA	14.6$\pm$0.6	21.05	-9.1	-8.2	-2.7	2.48	2/5

Acknowledgment

This research is co-financed by Greece and the European Union (European Social Fund- ESF) through the Operational Programme "Human Resources Development, Education and Lifelong Learning" in the context of the project "Strengthening Human Resources Research Potential via Doctorate Research" (MIS-5000432), implemented by the State Scholarships Foundation (IKY).

References

[1] R. Martins, N. Lourenço, N. Horta, J. Yin, P.-I. 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," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 1, pp. 69–82, 2019.

[2] G. İslamoğlu, T. O. Çakici, E. Afacan, and G. Dündar, "Artificial neural network assisted analog ic sizing tool," in *2019 16th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, 2019, pp. 9–12.

[3] J. P. Rosa, D. J. Guerra, N. C. Horta, R. M. Martins, N. C. Lourenço *et al.*, *Using artificial neural networks for analog integrated circuit design automation*. Springer, 2020, vol. 1.

[4] W. Lyu, P. Xue, F. Yang, C. Yan, Z. Hong, X. Zeng, and D. Zhou, "An efficient bayesian optimization approach for automated optimization of analog circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 6, pp. 1954–1967, 2017.

[5] K. Touloupas and P. P. Sotiriadis, "Locomobo: A local constrained multi-objective bayesian optimization for analog circuit sizing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2021.

[6] K. Touloupas, N. Chouridis, and P. P. Sotiriadis, "Local bayesian optimization for analog circuit sizing," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*, 2021, pp. 1237–1242.

[7] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[8] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2, no. 3.

[9] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[10] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer methods in applied mechanics and engineering*, vol. 186, no. 2-4, pp. 311–338, 2000.

[11] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," *Advances in neural information processing systems*, vol. 20, 2007.

[12] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik, "Automatic chemical design using a data-driven continuous representation of molecules," *ACS central science*, vol. 4, no. 2, pp. 268–276, 2018.

[13] A. Tripp, E. Daxberger, and J. M. Hernández-Lobato, "Sample-efficient optimization in the latent space of deep generative models via weighted retraining," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11 259–11 272, 2020.

[14] M. Drakaki, A. A. Hatzopoulos, and S. Siskos, "Cmos inductor performance estimation using z- and s-parameters," in *2007 IEEE International Symposium on Circuits and Systems*, 2007, pp. 2256–2259.