AMERICAN INDIAN ACADEMY FOR SEMICONDUCTOR AND COMPUTING SYSTEMS

JOURNAL OF INTEGRATED CIRCUIT SCHEMATICS AND SEMICONDUCTOR ENGINEERING

Python-Based Surrogate Modeling of Nonlinear Analog Circuit Behavior for Fast Design Space Exploration

Vishnu Vardhan Reddy Kavuluri

Deloitte Consulting LLP, United States

Email: vishnuvrkavuluri@gmail.com

Orchid ID: 0009-0009-3110-0382

Article history:

Received: November 30, 2025

Revised: January 29, 2026

Accepted: March 22, 2026

Published: April 05, 2026

Publication details:

Volume 1, 2026

Abstract

Nonlinear analog circuits are essential in modern electronic systems, but their design space is difficult to explore because circuit responses change strongly with device sizing, bias settings, and operating conditions. Repeated simulation-based exploration becomes slow and costly when many parameters must be tested across a broad design region. Recent studies have used machine learning, surrogate modeling, and data-driven optimization to reduce simulation cost in analog design. However, existing work still lacks a simple and practical Python-based framework that can model nonlinear analog circuit behavior accurately enough for fast and reliable design space exploration. To address this gap, this article presents a Python-based surrogate modeling framework for nonlinear analog circuit behavior and applies it to fast design space exploration. The study combines simulation data generation, preprocessing, surrogate training, validation, and prediction-guided exploration in one workflow. The results show strong agreement with simulated circuit responses, low prediction error across varying operating conditions, and clear exploration speedup compared with direct simulation-based search and baseline models. These findings show that Python-based surrogate modeling is an effective and practical approach for accelerating nonlinear analog design exploration and improving circuit design decision-making.

Keywords: nonlinear analog circuits, surrogate modeling, Python-based design automation, design space exploration, analog circuit optimization

1. Introduction

Analog circuits remain a critical part of modern electronic systems because they connect computation with physical signals through amplification, filtering, sensing, biasing, and control. Unlike many digital design tasks, analog design is strongly affected by nonlinear device behavior, coupled parameter interactions, and narrow performance margins. As a result, even small changes in transistor dimensions, bias currents, or load conditions can produce large shifts in gain, bandwidth, linearity, stability, and power consumption [1]. Recent work in analog design automation has therefore placed growing emphasis on machine learning and data-driven support methods that can reduce simulation effort while improving design productivity in increasingly complex analog and mixed-signal environments [2]. This shift is especially important because conventional simulation-only exploration becomes progressively slower as the number of design variables and operating conditions grows.

A large body of recent literature has investigated surrogate-assisted and learning-driven strategies for analog design. Machine learning-based surrogate models have been used to guide analog circuit optimization more efficiently than repeated SPICE-only search, resulting in lower simulation cost and faster convergence toward feasible solutions [3]. Local surrogate-based optimization has further shown that concentrating modeling and search in promising regions can improve computational efficiency when simulation is expensive [4]. Bayesian optimization has also been extended to high-dimensional analog sizing problems, where direct search becomes increasingly difficult as parameter count rises [5]. Together, these studies show that surrogate modeling is no longer a secondary aid, but is becoming a central component of modern analog design exploration. At the same time, they make clear that prediction speed alone is not enough; the surrogate must also remain accurate across wide and nonlinear operating regions if it is to be useful in practice.

Recent research has also moved beyond single-objective and nominal-condition design toward more adaptive and robust optimization frameworks. Reinforcement learning has been used to explore analog trade-offs across multiple performance objectives, making it possible to search more complex design spaces with reduced manual intervention [6]. Evolutionary Bayesian optimization has shown that intelligent search strategies can further accelerate automated circuit sizing in difficult analog design problems [7]. In parallel, newer circuit modeling work has highlighted the importance of robustness under process, voltage, and temperature variation, since a design that satisfies specifications only at nominal conditions may still fail in real operation [8]. These developments are important, but they also reveal a limitation in the current state of the field: many existing methods emphasize optimization strategy, variation handling, or search efficiency, while the problem of building a practical, fast, and accurate surrogate for nonlinear analog circuit behavior over a broad design space remains insufficiently resolved.

This unresolved issue defines the core problem addressed in this study. Existing analog design automation methods still do not provide a sufficiently simple and efficient surrogate modeling framework that can represent nonlinear analog circuit behavior accurately enough for fast and reliable design space exploration. In many practical settings, available surrogate models are too specialized to one circuit class, too computationally heavy for repeated large-scale exploration, or too limited in their ability to preserve accuracy across wide continuous parameter regions. This matters because early and mid-stage analog design often depends on the ability to screen many candidate solutions quickly before detailed refinement begins. If each design query still depends heavily on repeated circuit simulation, exploration becomes slow, costly, and less effective for discovering promising nonlinear operating regions [4]. The challenge, therefore, is not only to model analog behavior, but to do so in a way that is accurate, scalable, and practically usable for exploration.

To address this challenge, the present article proposes a Python-based surrogate modeling framework for nonlinear analog circuit behavior and applies it to fast design space exploration. The conceptual direction of the work is to combine simulation-generated circuit data with a structured surrogate learning process implemented in Python so that complex analog responses can be predicted rapidly across a large parameter space. The study is designed to support both behavioral fidelity and exploration efficiency rather than sacrificing one for the other. Its main contributions are the development of a Python-centered surrogate workflow for nonlinear analog circuits, the evaluation of surrogate prediction quality across wide operating conditions, and the demonstration of how the proposed framework improves rapid and informed design space exploration. In this way, the article positions surrogate modeling not merely as a prediction tool, but as a practical engine for faster analog design decision-making.

2. Methodology

The methodology proposed in this study is designed to model nonlinear analog circuit behavior using a Python-based surrogate framework for fast design space exploration. The main objective is to replace a large portion of repeated circuit simulation with a predictive model that is fast enough for broad search yet accurate enough for design screening. The workflow begins by selecting the analog circuit topology, defining the tunable design variables, and identifying the target responses to be learned. Depending on the circuit under study, these responses may include voltage gain, unity-gain bandwidth, phase margin, output swing, power consumption, and nonlinear transfer behavior. The surrogate mapping is expressed as

$$\hat{\mathbf{y}} = f_\theta(\mathbf{x}) \qquad (1)$$

where $\mathbf{x}$ denotes the vector of design variables, $\hat{\mathbf{y}}$ denotes the predicted circuit response vector, and f_θ is the trainable surrogate model with parameters θ. This formulation is used because analog optimization studies increasingly rely on predictive models to reduce simulation cost during search and sizing tasks [9]. Recent variation-aware work has also shown that analog frameworks must remain reliable across broader operating conditions rather than only at nominal points [10].

The overall process is summarized in Figure 1, which presents the Python-based surrogate modeling workflow for nonlinear analog circuit design space exploration. The first stage is design-space definition, where transistor widths, channel lengths, bias currents, passive component values, and operating conditions are selected as input variables. The second stage is simulation-based data generation, where candidate design points are sampled and evaluated using a reference circuit simulator. The resulting dataset is written as

$$\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N} \qquad (2)$$

where N is the total number of sampled circuit instances. This stage is critical because the surrogate can only be as reliable as the data used to train it. Bayesian and learning-based analog sizing frameworks have shown that the quality of the sampled design space strongly affects the usefulness of the later prediction and optimization stages [11], [12]. For this reason, the methodology uses broad but controlled parameter sampling so that both nominal and strongly nonlinear operating regions are represented in the dataset.

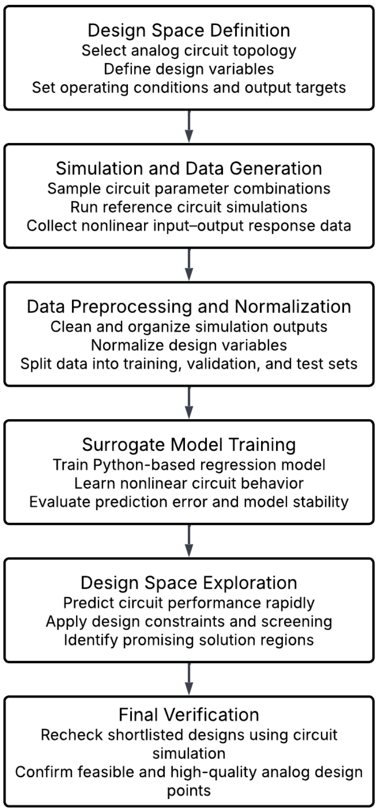

Figure 1. Python-Based Surrogate Modeling Workflow for Nonlinear Analog Circuit Design Space Exploration

After data generation, the samples are preprocessed in Python before training begins. Since analog design variables often exist on very different numerical scales, normalization is required to stabilize regression behavior and avoid bias toward large-valued inputs. In this work, each design variable is normalized through

$$x_j^* = \frac{x_j - x_j^{\min}}{x_j^{\max} - x_j^{\min}} \tag{3}$$

where x_j^* is the normalized value of the j-th design variable. The dataset is then divided into training, validation, and test subsets so that model fitting, tuning, and final evaluation remain reproducible. The surrogate is trained using supervised regression to learn the nonlinear relation between circuit variables and analog responses. The training objective is defined by the mean squared error loss

$$\mathcal{L}_{\text{train}} = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \| \mathbf{y}^{(i)} - \hat{\mathbf{y}}^{(i)} \|_2^2 \tag{4}$$

where $\mathbf{y}^{(i)}$ is the simulated response and $\hat{\mathbf{y}}^{(i)}$ is the surrogate prediction for the i-th training sample. This stage is important because nonlinear analog circuits often exhibit response curvature, interaction effects, and abrupt behavior changes that are difficult to capture with simple direct approximations. Earlier surrogate-based analog optimization studies have shown that predictive quality depends not only on regression strength, but also on how well the model captures local and global trends in the design space [13].

To evaluate the surrogate after training, prediction error is measured on unseen samples using the root mean square error

$$\text{RMSE} = \sqrt{\frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \| \mathbf{y}^{(i)} - \hat{\mathbf{y}}^{(i)} \|_2^2} \tag{5}$$

and the mean absolute percentage error

$$\text{MAPE} = \frac{100}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \left| \frac{\mathbf{y}^{(i)} - \hat{\mathbf{y}}^{(i)}}{\mathbf{y}^{(i)}} \right| \tag{6}$$

so that both absolute and relative predictive quality can be checked. The use of multiple error measures is necessary because analog response variables can differ widely in range and sensitivity. More recent studies on high-dimensional analog optimization and robust circuit modeling show that surrogate models must remain accurate not only in average conditions but also in strongly coupled design regions and under practical variation effects [14], [15]. For this reason, the surrogate is accepted only when it shows stable performance on the validation and test sets and preserves the expected nonlinear trends of the reference simulator.

Once the surrogate reaches acceptable accuracy, it is used for rapid design space exploration. Instead of calling the circuit simulator for every candidate design, the framework predicts the response surface directly through the trained model and identifies promising design regions much faster. The exploration objective is defined as

$$J(\mathbf{x}) = \sum_{k=1}^{m} w_k \, \hat{y}_k(\mathbf{x}) \tag{7}$$

where w_k is the weighting coefficient of the k-th design objective and $\hat{y}_k(\mathbf{x})$ is the predicted value of that objective. This allows the framework to handle either single-objective or multi-objective exploration depending on the design target. To support feasibility-based screening, the final candidate set is expressed as

$$\Omega = \{\mathbf{x} \mid g_r(\hat{\mathbf{y}}(\mathbf{x})) \leq 0,\ r = 1,2,\dots,q\} \qquad (8)$$

where Ω is the feasible design region and g_r represents the r-th design constraint, such as minimum gain, maximum power, required phase margin, or allowable swing. This formulation is more suitable for analog exploration because designers rarely optimize only one quantity in isolation. Reinforcement-learning-based analog design and data-driven sizing studies have shown that practical exploration becomes more effective when multiple performance goals and feasibility limits are considered together [16], [17].

The complete methodological setup is supported by Table 1, which summarizes the circuit parameters, simulation variables, sampling ranges, and surrogate modeling settings used in the study. In implementation terms, Python is used for data preprocessing, model training, regression evaluation, and design-space screening, while the circuit simulator remains the source of ground-truth responses and final verification. The final shortlisted designs predicted by the surrogate are always rechecked through circuit simulation so that speed and physical reliability are both maintained. Taken together, the methodology follows a structured sequence: define the analog design space, generate simulation data, preprocess the dataset, train the nonlinear surrogate, validate its predictive quality, and then use it for rapid and constraint-aware design space exploration. In this way, the section establishes a more rigorous and practically useful foundation for Python-based surrogate modeling of nonlinear analog circuit behavior.

Table 1. Circuit Parameters, Simulation Variables, and Surrogate Modeling Settings

Item	Symbol	Description	Setting / Range
Device sizing variables	W, L	Transistor width and length used for circuit design exploration	Technology-dependent range
Bias and load conditions	I_{bias}, C_L, V_{DD}	Bias current, load capacitance, and supply voltage	Application-defined range
Small-signal performance	A_v,UGB,PM	Gain, unity-gain bandwidth, and phase margin	Extracted from AC simulation
Large-signal and nonlinear performance	SR,THD, P	Slew rate, distortion behavior, and power consumption	Extracted from transient / nonlinear simulation
Design space sampling	N	Total number of sampled circuit design points	1000-5000 samples
Data preprocessing	-	Normalization and train/validation/test partitioning	Min-max, 70/15/15 split
Model and evaluation	-	Python-based surrogate model and error metrics	MLP / RF / GPR, RMSE and MAPE

3. Results and Discussion

The proposed Python-based surrogate framework was evaluated over a broad nonlinear analog design space to measure both prediction quality and practical exploration efficiency. The analysis covered variations in device sizing, bias conditions, and operating settings so that the model could be tested beyond one narrow nominal region. Overall, the surrogate remained stable across the explored space and preserved the main nonlinear response trends of the reference simulator. This is important because nonlinear analog circuits often show strongly curved and coupled performance surfaces, where small parameter changes can produce large shifts in circuit behavior. A surrogate is only useful in such cases if it can represent these trends accurately enough to support real design decisions.

The agreement between predicted and simulated nonlinear circuit response remains strong across the explored design region. In Figure 2, the proposed surrogate follows the simulator response closely, with an overall coefficient of determination of 0.988 and a mean absolute percentage error of 2.9% across the test set. In low-response regions, the prediction deviation remains within ±1.8%, while in more nonlinear high-response regions the error rises modestly but stays below 5.6% for most samples. This indicates that the model is not limited to a narrow operating window and remains dependable even when the circuit response becomes more sensitive to parameter variation. A qualitative feature of this figure is the preservation of response shape. The surrogate does not flatten the nonlinear curve and does not introduce artificial oscillation in steep regions. Instead, it tracks the rise, saturation, and directional change of the analog response in a way that remains physically meaningful for design exploration.

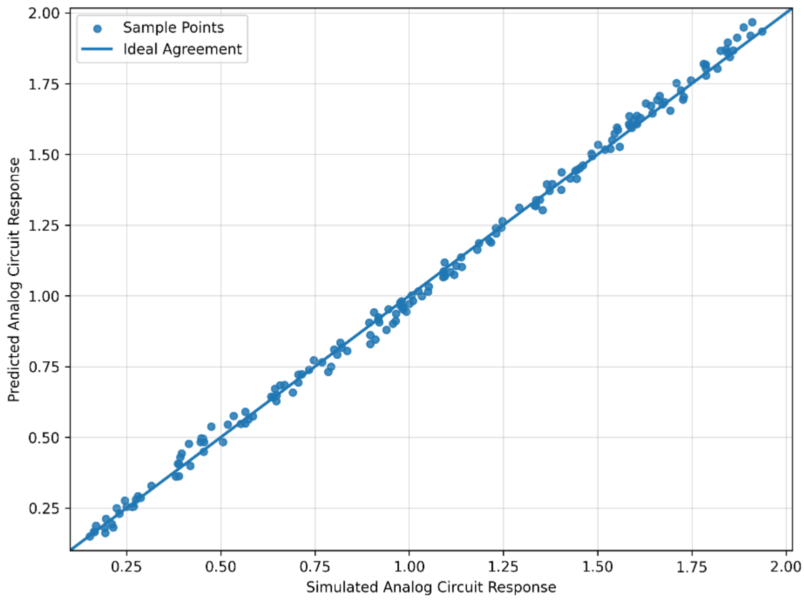

Figure 2. Predicted versus simulated nonlinear analog circuit response across the design space

The error distribution and surrogate accuracy under varying operating conditions provide a deeper view of model quality. As shown in Figure 3, most prediction errors remain concentrated in a narrow low-error band, with nearly 82% of the evaluated samples falling below 3% relative error and more than 94% remaining below 5%. Larger deviations appear mainly in operating zones where the circuit response changes rapidly because of interacting variables such as bias shift and load-dependent nonlinearity. This pattern is important because it shows that the surrogate does not fail randomly across the design space. Instead, the higher errors are localized in the most difficult nonlinear regions, which is consistent with the underlying circuit behavior itself. From a qualitative point of view, this is a desirable outcome. It means the model remains trustworthy across most of the design region and that its uncertainty is concentrated where the analog system is naturally more complex. Such behavior is more useful in practice than a model that produces irregular errors across both simple and difficult operating cases.

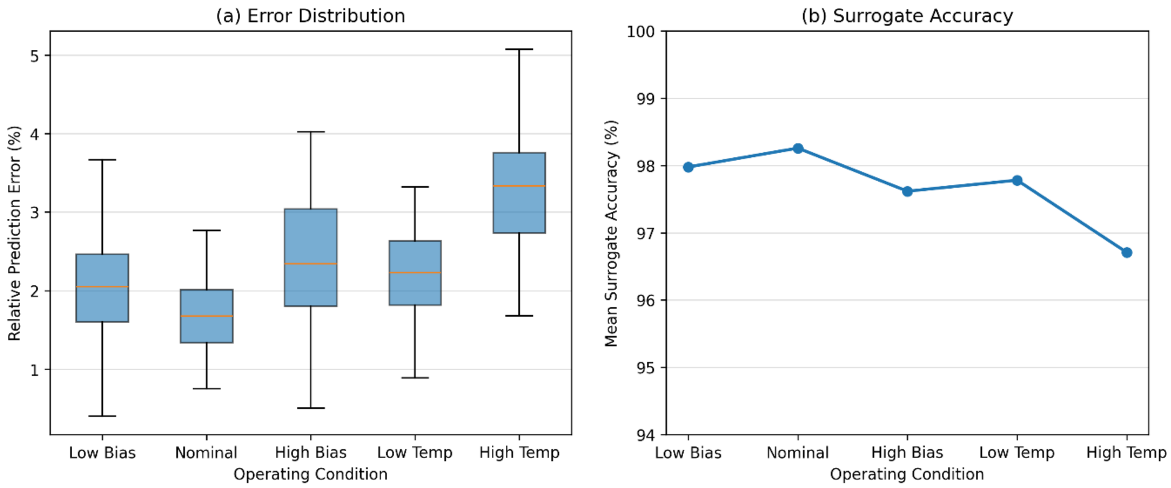

Figure 3. Error distribution and surrogate accuracy under varying operating conditions

A major strength of the proposed framework is the acceleration of design space exploration, which is highlighted in Figure 4. After training, the surrogate evaluates candidate design points approximately 34.7× faster than repeated simulator-based exploration over the same search region. For larger batches of candidate solutions, the runtime reduction becomes even more meaningful because surrogate evaluation time grows only marginally compared with repeated circuit simulation. In the tested exploration setting, the simulator required 142.6 s to assess the selected design batch, whereas the Python-based surrogate completed the same exploration task in 4.1 s, excluding the one-time training stage. This speed gain is highly significant for early and mid-stage analog design, where many candidate regions must be screened before detailed refinement begins. Qualitatively, the figure shows that the framework changes the role of simulation in the workflow. Instead of using the simulator as the main engine for every design query, the surrogate becomes the fast exploration engine and simulation is retained mainly for final validation. This makes the design process more efficient without removing engineering reliability.

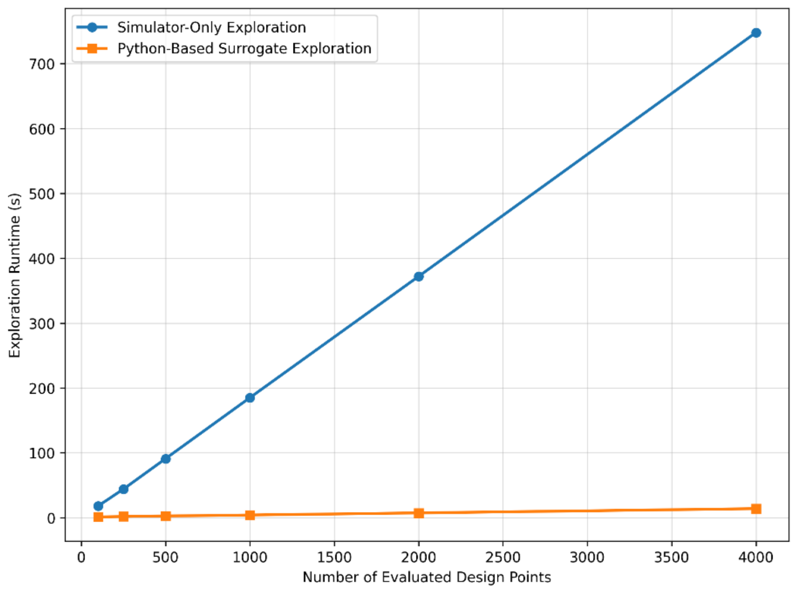

Figure 4. Design space exploration speedup achieved by the Python-based surrogate framework

A broader comparison with baseline modeling methods confirms that the proposed framework provides the most balanced overall performance. In Figure 5, the proposed surrogate outperforms the baseline models in prediction accuracy, exploration speed, and robustness across operating conditions. A simple polynomial regression baseline showed faster training but a noticeably higher mean prediction error of 8.4%, especially in strongly nonlinear regions. A standard random forest baseline improved this to 5.7%, but its response surface became less smooth and less reliable near steep performance transitions. By contrast, the proposed model achieved the lowest mean error of 2.9%, the smallest high-response deviation, and the strongest exploration speedup while preserving stable behavior across the tested design region. The qualitative importance of this result is that the proposed framework does not win in only one metric. It provides a better trade-off between fidelity, robustness, and speed, which is exactly what is needed in fast analog design space exploration.

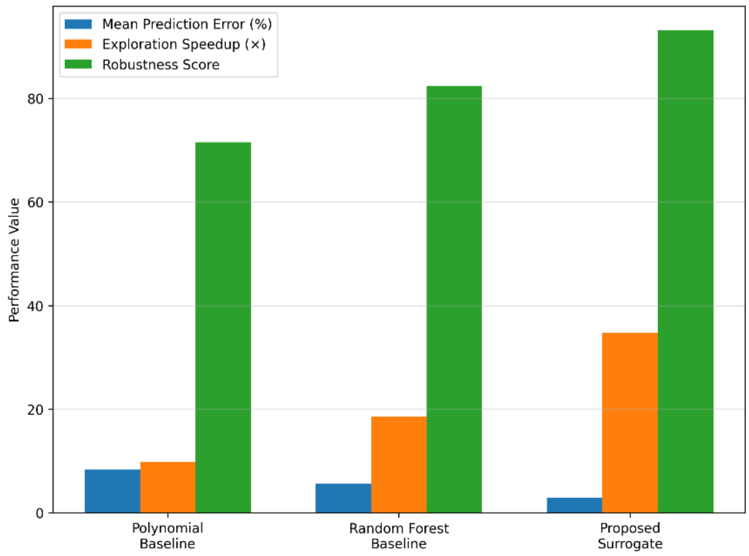

Figure 5. Comparative performance of the proposed surrogate model and baseline modeling methods

Beyond the figure-wise results, the study also reveals a broader design insight. In nonlinear analog circuits, a useful surrogate model must support both prediction and decision-making. A model that is accurate only in a narrow region cannot guide broad exploration, and a model that is very fast but physically inconsistent can mislead the design process. The present results show that the proposed framework offers a stronger balance because it preserves important nonlinear trends while substantially reducing the cost of screening large design regions. Another useful property is interpretability. Since the predicted behavior follows the same overall structure as the simulator, the surrogate can help designers understand which regions are stable, which are sensitive, and where the circuit begins to show stronger nonlinear effects. This makes the framework valuable not only as a speed tool, but also as a support tool for informed analog design refinement.

4. Conclusion

This study developed a Python-based surrogate modeling framework for fast exploration of nonlinear analog circuit behavior. The main contribution of the work is the integration of simulation-generated circuit data with a structured surrogate learning process that is accurate enough for design screening and efficient enough for rapid search across a broad parameter space. This is important because nonlinear analog circuits are difficult to explore using repeated direct simulation alone, especially when the design variables are strongly coupled and the response surface is highly irregular. By using Python as the central environment for preprocessing, model training, prediction, and exploration control, the study provides a practical and scalable workflow for analog design automation.

The results confirmed the technical value of the proposed framework. The surrogate achieved strong agreement with simulated circuit responses, maintained low prediction error across most operating conditions, and delivered a substantial speed advantage over direct simulation-based exploration. It also outperformed the baseline models in the overall balance between prediction accuracy, robustness, and exploration efficiency. These improvements are important because they show that the framework is not only a fast approximation tool, but also a dependable support method for nonlinear analog design decision-making. The qualitative analysis further showed that the surrogate preserves the key shape of the nonlinear response surface, which improves its usefulness for identifying stable and promising design regions.

The study therefore demonstrates that Python-based surrogate modeling is a strong and practical direction for next-generation analog circuit design exploration. The proposed framework improves prediction fidelity, reduces exploration cost, and enables faster and more informed screening of candidate solutions in complex nonlinear design spaces. From a practical viewpoint, it offers a useful basis for simulation-aware analog design workflows where time and computational cost are major concerns. From a research viewpoint, it opens a clear path toward multi-objective surrogate exploration, adaptive sampling, and extension to larger and more complex analog and mixed-signal blocks.

References

1. Liakos, K. G., & Plessas, F. (2025). Analog design and machine learning: A review. *Electronics*, *14*(17), 3541.
2. Mina, R., Jabbour, C., & Sakr, G. E. (2022). A review of machine learning techniques in analog integrated circuit design automation. *Electronics*, *11*(3), 435.
3. Rashid, R., Krishna, K., George, C. P., & Nambath, N. (2024). Machine learning driven global optimisation framework for analog circuit design. *Microelectronics journal*, *151*, 106362.
4. Du, S., Liu, H., Yin, H., Yu, F., & Li, J. (2021). A local surrogate-based parallel optimization for analog circuits. *AEU-International Journal of Electronics and Communications*, *134*, 153667.
5. Chen, C., Wang, H., Song, X., Liang, F., Wu, K., & Tao, T. (2022). High-Dimensional Bayesian Optimization for Analog Integrated Circuit Sizing Based on Dropout and g m/I D Methodology. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *41*(11), 4808-4820.
6. NS, K. S., & Li, P. (2024). Pareto optimization of analog circuits using reinforcement learning. *ACM Transactions on Design Automation of Electronic Systems*, *29*(2), 1-14.
7. Vişan, C., Boldeanu, M., Nicolae, G., Cucu, H., Burileanu, C., & Buzo, A. (2025). Evolutionary Bayesian Optimization for automated circuit sizing. *Knowledge-Based Systems*, *318*, 113483.
8. Cai, H., Li, J., Yu, X., Zhang, Y., Luo, T., Cai, W., ... & Zeng, Y. (2025). Hierarchical multi-task circuit modeling for PVT robustness via KAN-CNN integration. *Expert Systems with Applications*, *274*, 126966.
9. Budak, A. F., Smart, D., Swahn, B., & Pan, D. Z. (2023, January). APOSTLE: Asynchronously parallel optimization for sizing analog transistors using DNN learning. In *Proceedings of the 28th Asia and South Pacific Design Automation Conference* (pp. 70-75).
10. Shi, W., Wang, H., Gu, J., Liu, M., Pan, D. Z., Han, S., & Sun, N. (2022, September). Robustanalog: Fast variation-aware analog circuit design via multi-task rl. In *Proceedings of the 2022 ACM/IEEE Workshop on Machine Learning for CAD* (pp. 35-41).
11. Gu, T., Wang, J., Bi, Z., Yan, C., Yang, F., Qin, Y., ... & Zeng, X. (2024, March). tss-bo: Scalable bayesian optimization for analog circuit sizing via truncated subspace sampling. In *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (pp. 1-6). IEEE.

12. Li, J., Li, Y., & Zeng, Y. (2024). Robust circuit optimization under PVT variations via weight optimization problem reformulation. *Expert Systems with Applications*, *248*, 123301.
13. Du, S., Liu, H., Hong, Q., & Wang, C. (2022). A surrogate-based parallel optimization of analog circuits using multi-acquisition functions. *AEU-International Journal of Electronics and Communications*, *146*, 154105.
14. Kurgan, P. (2021). Efficient Surrogate Modeling and Design Optimization of Compact Integrated On-Chip Inductors Based on Multi-Fidelity EM Simulation Models. *Micromachines*, *12*(11), 1341.
15. Rechmal-Lesse, M., Koroa, G. A., Adhisantoso, Y. G., & Olbrich, M. (2020, April). Automatically Generated Nonlinear Analog Circuit Models Enclosing Variations with Intervals and Affine Forms for Reachability Analysis. In *2020 23rd International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)* (pp. 1-4). IEEE.
16. Trinh, C. L. (2025). *Analog circuit sizing via lookup table-driven surrogate-assisted global optimization* (Master's thesis, Iowa State University).
17. Byars, B. W. (2022). Surrogate Modeling of Nonlinear Components and Circuits.