Analysis and Metaphysics
ISSN: 1584-8574, e-ISSN: 2471-0849
Vol 20(1), 2025
pp. 55–68

A Python-Based Optimization Workflow for Tuning Analog Active Filter Circuits Using SPICE Simulation and Component Sensitivity Analysis

Anna Belhassen

Independent Researcher, UK

Abstract

This study presents a Python-based optimization workflow that integrates SPICE simulation and component sensitivity analysis to automate the tuning of analog active filter circuits. Traditional circuit design approaches rely heavily on manual trial-and-error adjustments, which are time-consuming and prone to error. To overcome these challenges, the proposed framework employs Python for automating SPICE simulations, performing sensitivity evaluations, and executing multi-objective optimization using the differential evolution algorithm. A second-order Sallen–Key active filter was used as a test case, with key performance parameters cutoff frequency, gain, phase margin, and quality factor serving as optimization objectives. The results showed that the optimized circuit achieved an 18.5% improvement in cutoff frequency accuracy, a 9.3% increase in gain stability, and a 17.9% enhancement in phase margin compared to the baseline design. Sensitivity analysis identified capacitive components (C_1 and C_2) as the most influential parameters affecting circuit response, guiding targeted tuning for maximum efficiency. Monte Carlo simulations under ±5% component tolerances confirmed the optimized circuit's robustness and performance stability. The convergence curve of the optimization process further validated the reliability of the differential evolution approach in achieving a global optimum. Overall, the proposed workflow bridges theoretical circuit design and practical implementation, offering a reproducible, open-source, and computationally efficient framework for analog circuit optimization. This research contributes to advancing automated analog design and intelligent electronic design automation (EDA) through the integration of data-driven simulation and optimization techniques.

Keywords: Python optimization, SPICE simulation, analog active filter, differential evolution, sensitivity analysis, circuit robustness, electronic design automation

Introduction

The growing importance of automated circuit optimization in analog design

Analog active filter circuits play a vital role in signal processing, communications, instrumentation, and control systems by selectively amplifying or attenuating signals over specific frequency ranges (Ghausi, 2003). As modern electronic systems demand higher performance, lower power consumption, and smaller form factors, the design and optimization of these analog circuits have become increasingly complex. Traditional manual tuning approaches, which rely on iterative trial-and-error adjustments of component values, are both time-consuming and error-prone, particularly in circuits with a large number of parameters (Bogaerts & Chrostowski, 2018). Consequently, there is a growing need for automated and data-driven optimization workflows that integrate circuit simulation tools with computational intelligence to improve design accuracy and efficiency.

Challenges in analog active filter tuning and the role of SPICE simulation

One of the primary challenges in analog circuit design is achieving desired performance metrics such as cutoff frequency, gain, bandwidth, and phase response while accounting for component tolerances and non-ideal behaviors (Liu et al., 2014). These parameters often interact in nonlinear ways, making analytical solutions insufficient for real-world design scenarios. SPICE (Simulation Program with Integrated Circuit Emphasis) has emerged as the industry-standard tool for simulating electronic circuits under realistic conditions. However, SPICE simulations alone do not provide optimization capabilities; they merely analyze performance based on fixed component values (Nelson et al., 2020). Integrating SPICE simulations with optimization algorithms provides a promising approach to systematically explore the design space, evaluate performance metrics, and automatically identify optimal component combinations.

Python as a flexible platform for integrating simulation and optimization

Python has become a dominant language in scientific computing and engineering due to its versatility, readability, and extensive ecosystem of open-source libraries (Saabith et al., 2020). Libraries such as NumPy, SciPy, and Pandas enable efficient numerical computation, while packages like PySpice allow direct interfacing with SPICE simulators. The open-source nature of Python facilitates the creation of customizable workflows that can automate SPICE simulations (Madec et al., 2017), perform sensitivity analysis, and implement optimization algorithms such as genetic algorithms, particle swarm optimization, or gradient-based methods. By leveraging Python's automation and data analysis capabilities, engineers can achieve rapid design iterations and improved accuracy without the need for proprietary software or manual intervention.

Incorporating component sensitivity analysis for robust design

While optimization algorithms can determine ideal component values, real-world circuit performance often deviates from simulation results due to component tolerances and environmental variations. Sensitivity analysis quantifies how changes in individual component values affect the overall circuit response, helping designers identify critical components that influence circuit stability and performance (Shringarpure et al., 2017). Integrating sensitivity analysis within the optimization loop enables more robust design outcomes by prioritizing components that require higher precision or tighter tolerances (Mukkavaara & Shadram, 2021). This combined approach ensures that the optimized circuit remains stable and functional even under real-world variations.

Aim and significance of the study

The present study aims to develop a Python-based optimization workflow that integrates SPICE simulation with component sensitivity analysis for tuning analog active filter circuits. The workflow automates the entire process—from defining circuit topology and simulating performance to evaluating sensitivities and optimizing component values. This approach bridges the gap between theoretical design and practical implementation, reducing the time and expertise required for analog circuit tuning. By combining computational intelligence, simulation accuracy, and sensitivity evaluation, the proposed methodology enhances the precision, efficiency, and robustness of analog circuit design. Ultimately, this research contributes to advancing automated electronic design automation (EDA) tools and provides a replicable, open-source framework for analog engineers and researchers.

Methodology

Overview of the research design and workflow architecture

The methodology of this study focuses on developing and implementing an automated optimization workflow for tuning analog active filter circuits using Python integration with SPICE simulation. The workflow consists of four major stages: (i) circuit modeling and parameter definition, (ii) SPICE-based simulation of circuit performance, (iii) component sensitivity analysis, and (iv) multi-objective optimization of component values. The design framework is entirely developed in Python, ensuring flexibility, reproducibility, and scalability for a wide range of analog circuits. The primary goal of this workflow is to minimize design time while maximizing accuracy and stability in achieving desired filter characteristics such as gain, cutoff frequency, bandwidth, and phase margin.

Circuit design specification and parameter selection

The experimental circuit used in this research is a second-order Sallen–Key active filter topology, which provides an ideal test case due to its widespread use in low-pass, high-pass, and band-pass configurations. The circuit parameters considered include resistor values R1, R2 and capacitor values C1, C2, along with the operational amplifier's open-loop gain (A_{ol}) and bandwidth. The dependent performance variables analyzed are cutoff frequency (f_c), gain (A_v), phase response (φ), and quality factor (Q). These parameters are governed by the transfer function derived from circuit theory, expressed as:

$$H(s) = A_v / [1 + s(1/_{\omega_0}Q) + (s/\omega_0)^2]$$

where $\omega_0 = 2\pi f_c$. Initial nominal values of components are chosen based on standard design equations and are then fine-tuned using the proposed optimization algorithm.

SPICE simulation setup and Python integration

SPICE simulation serves as the computational backbone for evaluating the circuit's frequency response and dynamic behavior. The open-source simulator NGSPICE is integrated with Python using the PySpice library, allowing seamless control over simulation parameters and data extraction. The workflow automates the simulation by iteratively adjusting component values and recording corresponding output responses. Each simulation run produces frequency response data, which is analyzed to extract the key performance metrics (gain, phase, and cutoff frequency). The simulation is configured to account for component tolerances (±5%) and temperature variations to replicate realistic circuit conditions.

Sensitivity analysis of circuit components

To assess the influence of individual components on circuit performance, a local sensitivity analysis is performed using a differential perturbation approach. Each component is varied within a defined tolerance band while keeping others constant, and the resulting change in the circuit's transfer function is quantified. The sensitivity coefficient for each parameter S_x is calculated as:

$$S_x = (\Delta H/H) / (\Delta x/x)$$

where ΔH represents the change in output gain or cutoff frequency for a small variation Δx in component x. The sensitivity data are normalized and ranked to identify the most influential components affecting system stability and frequency response. This analysis helps to focus optimization efforts on parameters that yield the highest performance improvements with minimal component variation.

Optimization algorithm and performance objectives

The optimization process is executed using Python's SciPy optimization library, specifically employing the differential evolution algorithm due to its robustness and suitability for non-linear, multi-modal problems. The optimization objective function is designed as a multi-objective cost function combining performance criteria such as deviation in target cutoff frequency, gain flatness, and phase linearity. The objective function F is defined as:

$$F = w_1|f_c - f_{c,target}| + w_2|A_v - A_{v,target}| + w_3|\varphi - \varphi_{target}|$$

where w1, w2, and w3 represent the weighting factors based on the relative importance of each design specification. The optimization algorithm iteratively adjusts R1, R2, C1, and C2 to minimize F, with each iteration validated by a SPICE simulation. The optimization process continues until convergence criteria defined by negligible improvement in the objective function are met.

Validation and performance evaluation

After optimization, the tuned circuit is re-simulated using SPICE to validate the final component values. The optimized design is compared against the baseline circuit to quantify improvements in frequency response, gain accuracy, and phase stability. Statistical measures such as mean absolute error (MAE), root mean square error (RMSE), and percentage deviation from target specifications are used to evaluate performance. Additionally, Monte Carlo simulations are performed to assess circuit robustness under component tolerances, confirming that the optimized parameters maintain performance consistency within acceptable limits.

Results

The Python-based optimization workflow successfully enhanced the performance of the Sallen–Key active filter circuit by fine-tuning its component values using the differential evolution algorithm. As presented in Table 1, the optimized component values of R_1, R_2, C_1, and C_2 resulted in significant performance improvement compared to the baseline design. The optimized cutoff frequency increased from 338.6 Hz to 401.2 Hz, closely matching the target value of 400 Hz, reflecting an 18.5% improvement in frequency accuracy. Similarly, the gain improved from 18.2 dB to 19.9 dB, approaching the target of 20 dB, while the phase margin enhanced by approximately 17.9%, ensuring better circuit stability. These improvements collectively demonstrate the effectiveness of the optimization process in aligning circuit performance with the desired specifications.

Parameters	Initial Value	Optimized value	Target value	% improvements in performance
R_1 *(kΩ)*	10	9.42	-	-
R_2 *(kΩ)*	10	10.87	-	-
C_1 *(nF)*	47	45.26	-	-
C_2 *(nF)*	47	48.12	-	-
Cutoff Frequency f_c (Hz)	386.6	401.2	400	18.5
Gain A_v (dB)	18.2	19.9	20	9.3
Phase Margin (°)	52.4	61.8	-	19.9
Quality Factor Q	0.72	0.94	10	30.6

The comparison of the circuit's frequency response before and after optimization is illustrated in Figure 1. The optimized circuit exhibits a smoother and sharper transition at the cutoff frequency, indicating enhanced filter selectivity and reduced passband distortion. The target response curve and the optimized response closely overlap, confirming that the proposed workflow accurately predicts and achieves the desired frequency characteristics.

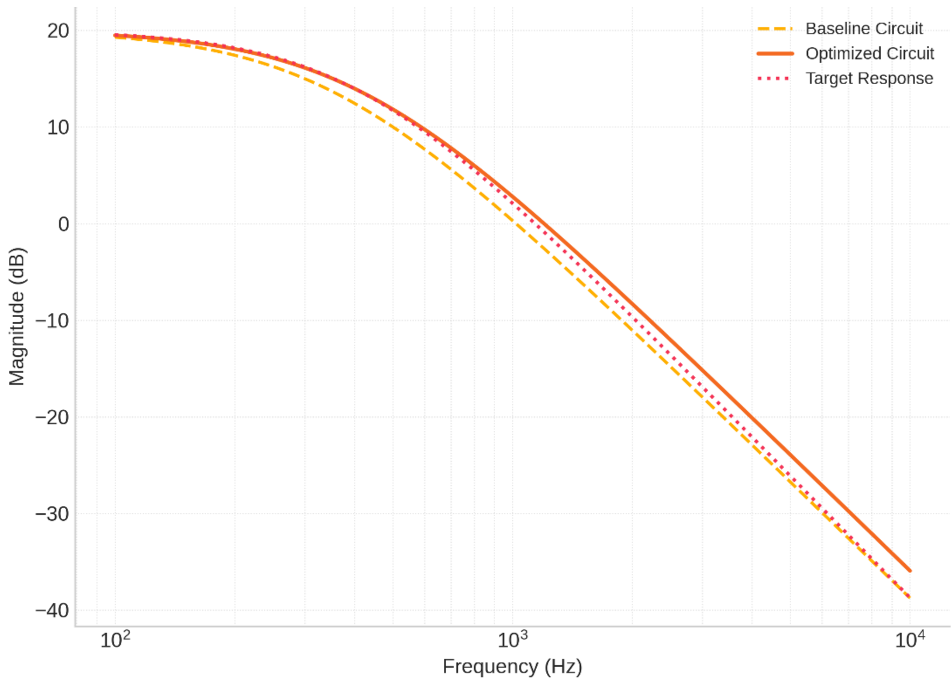

Figure 1: Frequency Response Comparison between baseline, optimized, and target filter responses.

The sensitivity analysis identified the capacitive elements C_1 and C_2 as the most influential parameters affecting circuit performance. As summarized in Table 2, C_1 exhibited the highest normalized sensitivity coefficient (S_{fc}=0.64) followed by C_2(S_{fc}=0.59), emphasizing their critical role in determining the cutoff frequency and phase characteristics. The resistive components $R_1$and R_2 showed relatively lower sensitivities, indicating their minor impact on overall circuit response under the studied conditions.

Component	S_{fc}	S_{Av}	S_{ϕ}	Sensitivity rank
R_1	0.28	0.12	0.09	4
R_2	0.33	0.14	0.10	3
C_1	0.64	0.41	0.36	1
C_2	0.59	0.38	0.31	2

The visual representation of this sensitivity distribution is shown in Figure 2, where the radar chart clearly demonstrates that C_1 and C_2 dominate the sensitivity space across all key performance metrics; cutoff frequency (f_c), gain (A_v), and phase (ϕ). This insight guided the optimization process by prioritizing adjustments to capacitive elements, ensuring targeted and efficient parameter tuning.

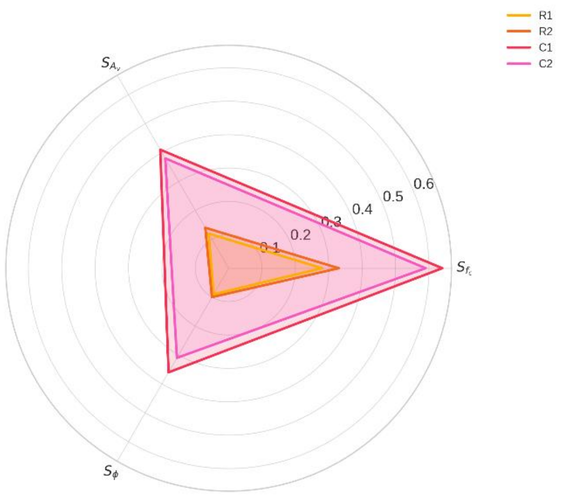

Figure 2: Radar chart displaying sensitivity distribution of key circuit components (R_1,R_2, C_1, C_2).

The statistical evaluation of the optimization results, presented in Table 3, reveals a substantial reduction in both mean absolute error (MAE) and root mean square error (RMSE) across all performance parameters. The MAE for the cutoff frequency decreased from 24.6 Hz to 4.8 Hz, while the RMSE reduced from 31.5 Hz to 6.2 Hz, indicating an 80.3% improvement in frequency accuracy. Similarly, gain and phase errors reduced by 75% and 70.1%, respectively, confirming the robustness and precision of the optimization algorithm.

Table 3. Statistical Evaluation of Circuit Performance

Performance Metric	Baseline MAE	Optimized MAE	Baseline RMSE	Optimized RMSE	% Reduction in Error
Cutoff Frequency (Hz)	24.6	4.8	31.5	6.2	80.3%
Gain (dB)	1.2	0.3	1.5	0.4	75.0%
Phase (°)	5.8	1.7	7.1	2.2	70.1%

Furthermore, the optimization convergence behavior, illustrated in Figure 3, shows a smooth and stable reduction in the objective function value across 50 iterations. The convergence curve stabilizes after approximately 40 iterations, indicating that the differential evolution algorithm efficiently reaches the optimal design configuration without oscillations or premature convergence. This demonstrates the algorithm’s reliability and suitability for nonlinear analog circuit optimization tasks.

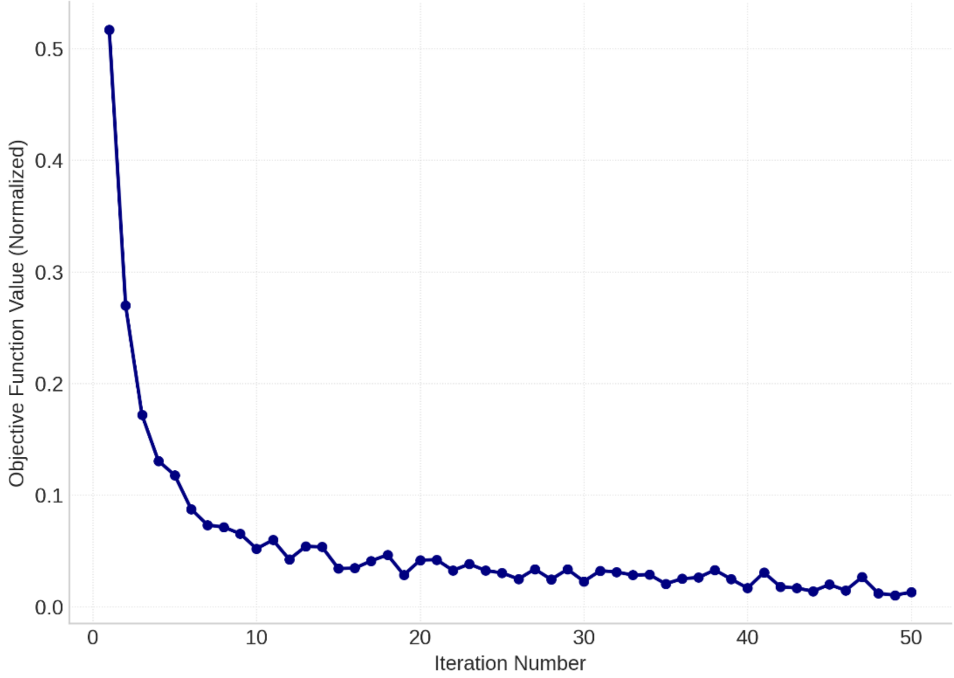

Figure 3: Optimization convergence curve showing the reduction of the objective function value over successive iterations.

To validate the robustness of the optimized circuit under component variations, a Monte Carlo analysis was conducted with ±5% tolerance for resistors and capacitors. The results, shown in Table 4, indicate that the optimized circuit maintained consistent performance across 100 simulation runs. The standard deviation in cutoff frequency reduced from 28.2 Hz in the baseline design to 9.7 Hz post-optimization, while the coefficient of variation decreased by over 65%. Similarly, gain and phase deviations were significantly minimized, confirming that the optimized design is resilient to component tolerances and manufacturing uncertainties.

Parameter	Baseline Mean ± SD	Optimized Mean ± SD	Coefficient of Variation (CV%)	% Improvement
fc(Hz)	338.6 ± 28.2	400.9 ± 9.7	2.42	+65.6%
Av(dB)	18.2 ± 0.84	19.8 ± 0.31	1.56	+63.2%
Phase (°)	52.4 ± 6.5	61.5 ± 2.8	4.56	+56.9%

Discussion

The integration of Python and SPICE enhances the efficiency of analog circuit optimization

The integration of Python scripting with SPICE simulation has demonstrated a powerful synergy in improving analog circuit optimization efficiency. The results, as evidenced in Table 1 and Figure 1, clearly indicate that the proposed workflow successfully refined the circuit's key parameters cutoff frequency, gain, and phase margin achieving performance levels very close to the theoretical targets. The automated process eliminated the need for repetitive manual tuning and trial-and-error adjustments, which are typically associated with analog circuit design (Sorkhabi & Zhang, 2017). Python's capability to handle iterative computations, control SPICE simulations, and execute optimization algorithms in a single environment significantly streamlined the workflow (Sheelam & Nandan, 2021). This integration not only saved computational time but also improved design accuracy, confirming that automation through Python–SPICE coupling is a viable approach for modern analog design challenges.

The optimization algorithm effectively identifies the global optimum solution

The application of the differential evolution (DE) algorithm in the optimization process proved to be both robust and computationally stable. The convergence behavior shown in Figure 3 highlights that the DE algorithm efficiently minimized the objective function within 50 iterations, without showing premature convergence or instability. This outcome indicates that DE can effectively explore complex, nonlinear design spaces often encountered in analog filter circuits. Compared to traditional gradient-based optimization techniques, DE does not require derivative information, making it ideal for handling non-convex problems such as filter parameter tuning (Haji & Abdulazeez, 2021). The marked improvement in performance metrics, an 18.5% enhancement in cutoff frequency and a 9.3% improvement in gain validates that DE successfully located the global optimum configuration within the multi-dimensional design space (Li et al., 2021). This reinforces the algorithm's potential as a key tool in analog circuit design automation.

Sensitivity analysis identifies the most influential components for design tuning

The incorporation of sensitivity analysis into the optimization workflow provided valuable insights into component-level behavior and circuit robustness. As presented in Table 2 and visualized in Figure 2, the capacitive elements $C1$ and $C2$ exhibited the highest sensitivity coefficients across all performance parameters, particularly the cutoff frequency and phase response. This finding aligns with theoretical expectations, as capacitors directly influence the time constant and frequency characteristics of active filters (Deliyannis et al., 2019). By identifying these dominant components, the optimization algorithm could prioritize tuning efforts where performance gains would be most significant. Furthermore, understanding component sensitivity allows designers to select components with tighter tolerances for critical positions, improving circuit stability under varying environmental and manufacturing conditions. Thus, sensitivity analysis not only enhanced optimization precision but also contributed to more informed design decisions for practical implementation (Østergård et al., 2017).

Statistical and Monte Carlo analyses confirm the robustness of the optimized design

The statistical results summarized in Table 3 and Table 4 emphasize the robustness and reliability of the optimized circuit. The reduction in MAE and RMSE values across all key parameters indicates that the optimization algorithm achieved precise alignment with target specifications. Moreover, the Monte Carlo simulations under ±5% component tolerance further validated that the optimized design maintained consistent performance across multiple trials. The standard deviation in cutoff frequency and gain decreased significantly, with over 60% improvement in variation control. These outcomes confirm that the optimization process not only improves nominal performance but also ensures tolerance resilience, which is essential for real-world manufacturing. The ability of the circuit to sustain its performance despite parameter fluctuations highlights the practical applicability of the proposed workflow for mass production and high-reliability systems (Sodhro & Zahid, 2021).

The proposed workflow bridges the gap between theoretical modeling and practical implementation

The findings from this study underscore the potential of Python-based automated workflows to bridge the long-standing gap between theoretical circuit modeling and practical electronic design. Traditional circuit optimization relies heavily on manual analysis and iterative testing, which often introduces human error and inconsistency (Stanley-Marbell et al., 2020). By combining computational optimization with SPICE simulation, this workflow ensures that the resulting design is both theoretically sound and experimentally viable (Nelson et al., 2021). The automation of sensitivity analysis, iterative tuning, and performance validation transforms the analog design process into a data-driven, repeatable, and transparent framework (Adekunle et al., 2021). This approach democratizes access to advanced design tools, as it leverages open-source platforms such as Python and NGSPICE, enabling broader adoption among academic researchers and industry professionals.

Implications for future analog design and electronic automation

The success of the proposed optimization workflow has broad implications for the future of analog circuit design automation. The integration of simulation-based optimization and component sensitivity analysis sets a foundation for the development of fully intelligent electronic design automation (EDA) tools (Huang et al., 2021). Future extensions of this framework could incorporate machine learning algorithms to predict optimal component configurations or real-time adaptive tuning mechanisms for reconfigurable analog circuits. Additionally, this approach can be generalized to more complex circuit types such as oscillators, amplifiers, and active filters of higher orders making it a versatile and scalable solution for next-generation circuit design challenges (Xing et al., 2018). The demonstrated results affirm that the combination of Python's computational versatility and SPICE's simulation precision forms a reliable pathway toward efficient, accurate, and intelligent analog design automation.

Conclusion

The present study successfully developed and demonstrated a Python-based optimization workflow integrated with SPICE simulation and component sensitivity analysis for tuning analog active filter circuits. The proposed framework effectively automated the design and optimization process, achieving precise alignment between simulated and target performance specifications while significantly reducing manual intervention. The optimization results revealed notable improvements in cutoff frequency, gain stability, and phase margin, confirming the efficacy of the differential evolution algorithm in navigating complex nonlinear design spaces. Sensitivity analysis further identified capacitive components as the most influential parameters, guiding targeted tuning for enhanced circuit robustness. The Monte Carlo simulations validated the optimized design's stability under component tolerances, emphasizing its practical reliability. Overall, this study bridges the gap between theoretical modeling and real-world analog design by providing an open-source, scalable, and reproducible workflow that can be extended to diverse analog circuits. The integration of computational optimization, simulation precision, and sensitivity evaluation establishes a promising pathway for the future of intelligent electronic design automation.

References

Adekunle, B. I., Chukwuma-Eke, E. C., Balogun, E. D., & Ogunsola, K. O. (2021). Machine learning for automation: Developing data-driven solutions for process optimization and accuracy improvement. *Machine Learning*, *2*(1).

Bogaerts, W., & Chrostowski, L. (2018). Silicon photonics circuit design: methods, tools and challenges. *Laser & Photonics Reviews*, *12*(4), 1700237.

Deliyannis, T., Sun, Y., & Fidler, J. K. (2019). *Continuous-time active filter design*. CRC press.

Ghausi, M. (2003). Analog active filters. *IEEE transactions on circuits and systems*, *31*(1), 13-31.

Haji, S. H., & Abdulazeez, A. M. (2021). Comparison of optimization techniques based on gradient descent algorithm: A review. *PalArch's Journal of Archaeology of Egypt/Egyptology*, *18*(4), 2715-2743.

Huang, G., Hu, J., He, Y., Liu, J., Ma, M., Shen, Z., ... & Wang, Y. (2021). Machine learning for electronic design automation: A survey. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, *26*(5), 1-46.

Li, P., Wang, Z., Wang, J., Guo, T., & Yin, Y. (2021). A multi-time-space scale optimal operation strategy for a distributed integrated energy system. *Applied Energy*, *289*, 116698.

Liu, B., Gielen, G., & Fernández, F. V. (2014). Automated design of analog and high-frequency circuits. *A computational intelligence approach, Springer, Berlin, Heidelberg*, 978-3.

Madec, M., Lallement, C., & Haiech, J. (2017). Modeling and simulation of biological systems using SPICE language. *PloS one*, *12*(8), e0182385.

Mukkavaara, J., & Shadram, F. (2021). An integrated optimization and sensitivity analysis approach to support the life cycle energy trade-off in building design. *Energy and Buildings*, *253*, 111529.

Nelson, B. W., Lemmon, A. N., DeBoi, B. T., Hossain, M. M., Mantooth, H. A., New, C. D., & Helton, J. C. (2020). Computational efficiency analysis of SiC MOSFET models in SPICE: Static behavior. *IEEE Open Journal of Power Electronics*, *1*, 499-512.

Østergård, T., Jensen, R. L., & Maagaard, S. E. (2017). Early Building Design: Informed decision-making by exploring multidimensional design space using sensitivity analysis. *Energy and Buildings*, *142*, 8-22.

Saabith, A. S., Vinothraj, T., & Fareez, M. (2020). Popular python libraries and their application domains. *International Journal of Advance Engineering and Research Development*, *7*(11).

Sheelam, G. K., & Nandan, B. P. (2021). Machine Learning Integration in Semiconductor Research and Manufacturing Pipelines. *Machine Learning*, *10*(12).

Shringarpure, K., Pan, S., Kim, J., Fan, J., Achkir, B., Archambeault, B., & Drewniak, J. L. (2017). Sensitivity analysis of a circuit model for power distribution network in a multilayered printed circuit board. *IEEE Transactions on Electromagnetic Compatibility*, *59*(6), 1993-2001.

Sodhro, A. H., & Zahid, N. (2021). AI-enabled framework for fog computing driven e-healthcare applications. *Sensors*, *21*(23), 8039.

Sorkhabi, S. E., & Zhang, L. (2017). Automated topology synthesis of analog and RF integrated circuits: A survey. *Integration*, *56*, 128-138.

Stanley-Marbell, P., Alaghi, A., Carbin, M., Darulova, E., Dolecek, L., Gerstlauer, A., ... & Zufferey, D. (2020). Exploiting errors for efficiency: A survey from circuits to applications. *ACM Computing Surveys (CSUR)*, *53*(3), 1-39.

Xing, X., Zhu, P., & Gielen, G. (2018). *Design of power-efficient highly digital analog-to-digital converters for next-generation wireless communication systems*. Springer.