Toward Automated Analog IC Design via Multi-Agent Framework with Foundation Model

Benchao Zhu
Fudan University
benchao-zhu@outlook.com

Yiheng Gu
Fudan University
kerwin.sdxknhg@gmail.com

Xiangtao Lu
Peking University
luxiangtao@stu.pku.edu.cn

***Abstract*—The front-end design of analog circuits, focusing on topology design and parameter tuning, presents substantial challenges. Even for widely used analog circuits such as operational amplifiers (opamps), the design process remains time-consuming and labor-intensive. To tackle the challenges, this paper proposes a multi-agent collaborative framework for automated operational amplifier design. The method leverages the synergy of three agents (Design Agent, Simulation Agent, and Analysis Agent) to achieve automated design and optimization of operational amplifiers. The Design Agent generates SPICE-compliant netlists, the Simulation Agent validates the netlists and extracts key performance metrics, and the Analysis Agent provides optimization suggestions based on simulation results. Combined with batch generation/screening and a normalized scoring method, the proposed framework efficiently generates circuit designs that meet performance requirements. Experimental results demonstrate that the method achieves the desired specifications in key metrics such as open-loop gain, gain bandwidth product, phase margin, slew rate, and power consumption, offering a novel solution for intelligent and automated analog circuit design.**

I. Introduction

Analog circuits serve as fundamental components in modern electronic systems, enabling critical functionalities ranging from portable electronics to Internet-of-Things applications through essential signal processing, data conversion, and communication interfaces. The design process for analog circuits presents inherent complexity, demanding synergistic integration of domain-specific physical intuition, mathematical analysis, and practical engineering expertise. Even for canonical structures like operational amplifiers (opamps), expert designers typically undergo laborious trial-and-error iterations involving meticulous parameter tuning and topology modifications to satisfy specifications.

Existing research in opamp design automation focuses primarily on two aspects: continuous parameter optimization (sizing) using black-box methods such as Bayesian optimization [1], [2] and reinforcement learning [3], [4], and discrete topology optimization employing genetic algorithms [5], Bayesian frameworks [6], or reinforcement learning [7]. However, these black-box approaches face fundamental limitations in handling the combinatorial topological design space and interpreting generated circuit structures, particularly due to their inability to incorporate domain knowledge spanning physics, mathematics, and engineering practice.

The emergence of large language models (LLMs) introduces new opportunities for Electronic Design Automation (EDA). While recent works demonstrate LLM capabilities in digital circuit design through RTL code generation and natural language interfaces, their application to analog circuits remains nascent. Two primary challenges hinder progress: (1) Semantic discontinuity between natural language descriptions and formal netlist representations of analog topologies, complicating structural-functionality mapping; (2) Data scarcity stemming from proprietary nature of expert-level analog designs and domain knowledge.

To overcome these limitations, this work proposes a multi-agent framework for automated analog circuit design. The methodology integrates three specialized agents: a Design Agent generating SPICE-compliant netlists, a Simulation Agent validating designs and extracting performance metrics, and an Analysis Agent providing optimization guidance. Through iterative batch generation, performance screening, and normalized scoring mechanisms, the framework successfully achieves target specifications for opamp designs. Experimental validation confirms the effectiveness of this approach in producing interpretable, human-acceptable analog circuit solutions.

The rest of the paper is organized as follows. Section II introduces opamp design problem. Section III describes the proposed framework. Section IV presents experimental results.

II. Problem Formulation

The operational amplifier design challenge constitutes a constrained multi-objective optimization problem. Formally, we define the design space as the Cartesian product of topological configurations $\mathcal{G}$ and parameter spaces $\mathcal{X}$. The mathematical formulation is expressed as:

$$\begin{aligned} \max_{g\in\mathcal{G},\mathbf{x}\in\mathcal{X}} \quad & f\left(g,\mathbf{x}\right)=\sum_{i=1}^{N_w} w_i\cdot\phi_i(s_i(g,\mathbf{x})) \\ \text{subject to} \quad & c_j(g,\mathbf{x})\geq c_{th}^j, \quad \forall j\in\{1,...,N_c\} \end{aligned} \tag{1}$$

where the objective function $f(\cdot)$ aggregates five normalized performance metrics ($N_w = 5$) through weighted summation: direct current gain s_1, gain-bandwidth product s_2, phase margin s_3, slew rate s_4, and power efficiency s_5. The normalization function $\phi_i(\cdot)$ maps each metric to [0,1] range.

The constraint set contains $N_c = 5$ essential specifications requiring strict satisfaction: AC gain (c_1), gain-bandwidth

Table I: Design Specification Thresholds

Metric	Symbol	Threshold
AC Gain	A_v	$> 60\,\mathrm{dB}$
Gain Bandwidth	GBW	$> 20\,\mathrm{MHz}$
Phase Margin	PM	$> 60^\circ$
Slew Rate	SR	$> 20\,\mathrm{V/\mu s}$
Static Current	I_{DD}	$< 500\,\mu\mathrm{A}$

product (c_2), phase margin (c_3), slew rate (c_4), and static current (c_5). The corresponding thresholds c_{th}^j are derived from the AICAS Grand Challenge specifications as enumerated in Table I.

III. Proposed Framework

A. Architectural Overview

The proposed framework establishes an automated circuit design system through synergistic multi-agent collaboration, employing three specialized agents that engage in iterative optimization cycles. As illustrated in Figure 1, the architecture embodies three fundamental design principles:

Conditional Co-Design Process: The tripartite agent system implements threshold-driven termination logic when design specifications are satisfied. The feedback loop terminates when all essential performance metrics simultaneously meet predefined thresholds:

$$\bigwedge_{i=1}^{n} (s_i \geq s_i^{th}) \tag{2}$$

where s_i represents the i-th performance metric (gain, bandwidth, etc.), and s_i^{th} denotes corresponding specification thresholds from Table I.

Stochastic Design Exploration: The framework leverages the inherent stochasticity in large language model generations to create solution diversity. Through multiple independent forward passes with temperature-scaled sampling, the LLM-based Design Agent produces variant netlist candidates:

$$p(\mathcal{N}_i|\mathcal{P}) = \prod_{t=1}^{T} \frac{\exp(z_t/\tau)}{\sum_{j=1}^{V} \exp(z_j/\tau)} \tag{3}$$

where τ controls the generation diversity through temperature scaling, $\mathcal{P}$ denotes the prompt context, and $\mathcal{N}_i$ represents the i-th candidate netlist. This probabilistic generation mechanism naturally induces solution space exploration without explicit parameter perturbation.

Self-Regulated Workflow Execution: Centralized orchestration through the AmpDesignController ensures robust process management with automated error recovery and traceability. The controller implements finite state machine logic:

$$s_{t+1} = f(s_t, a_t, o_t) \tag{4}$$

where s_t denotes system state at step t, a_t the agent action, and o_t the observation from simulation results.

B. Agent-Based Co-Design Methodology

The core innovation resides in the tripartite agent system depicted in Figure 2, where each agent exhibits specialized capabilities while maintaining contextual awareness through shared state information.

1) Design Agent (Generative Netlist Synthesis): The Design Agent implements constrained netlist generation through hybrid symbolic-numeric methods. Utilizing large language models with in-context learning capabilities, it produces SPICE-compliant circuit descriptions while enforcing process design rules. The generation process follows constrained beam search:

$$\mathcal{L} = \arg\max_{\theta} \sum_{i=1}^{N} \log P(y_i|x, \mathcal{C}) \tag{5}$$

where $\mathcal{C}$ represents design constraints encoded as prompt context. Unit consistency is enforced through dimensional analysis:

$$[W] = \mu m,\ [L] = \mu m,\ [I_{bias}] = \mu A \tag{6}$$

Parameter validation uses regular expressions with lookahead assertions to guarantee $0.2 \leq W \leq 100$ and $0.2 \leq L \leq 10$.

2) Simulation Agent (Numerical Verification): The Simulation Agent executes automated circuit characterization through standardized SPICE simulation protocols, utilizing competition-provided netlist templates to ensure consistency. Its core functionality encompasses three-phase analysis: AC frequency response evaluation for gain and phase margin determination, transient slew rate measurement under step input conditions, and DC operating point calculation for static current verification. Through systematic parsing of raw simulation outputs, the agent extracts quantitative performance metrics including gain-bandwidth product and power consumption, while concurrently implementing error diagnostics via regular expression matching on simulation log codes. Error conditions are classified according to predefined criticality levels, enabling efficient fault isolation during iterative refinement cycles.

3) Analysis Agent (Semantic Optimization): The Analysis Agent implements performance refinement through natural language processing of simulation outcomes. By synthesizing numerical results with historical design patterns, the LLM-based module generates contextual improvement suggestions through chain-of-thought prompting. The optimization process follows:

- **Critical Factor Identification**: Detects bottleneck metrics through comparative analysis of current/past simulation results
- **Instruction Synthesis**: Produces human-interpretable optimization directives (e.g., "Increase M1's W to $10\mu m$ for higher GBW")
- **Constraint Verification**: Ensures suggestions comply with process design rules before feedback integration

This approach enables topology-aware modifications without gradient computation, leveraging the LLM's parametric knowledge of analog design principles.

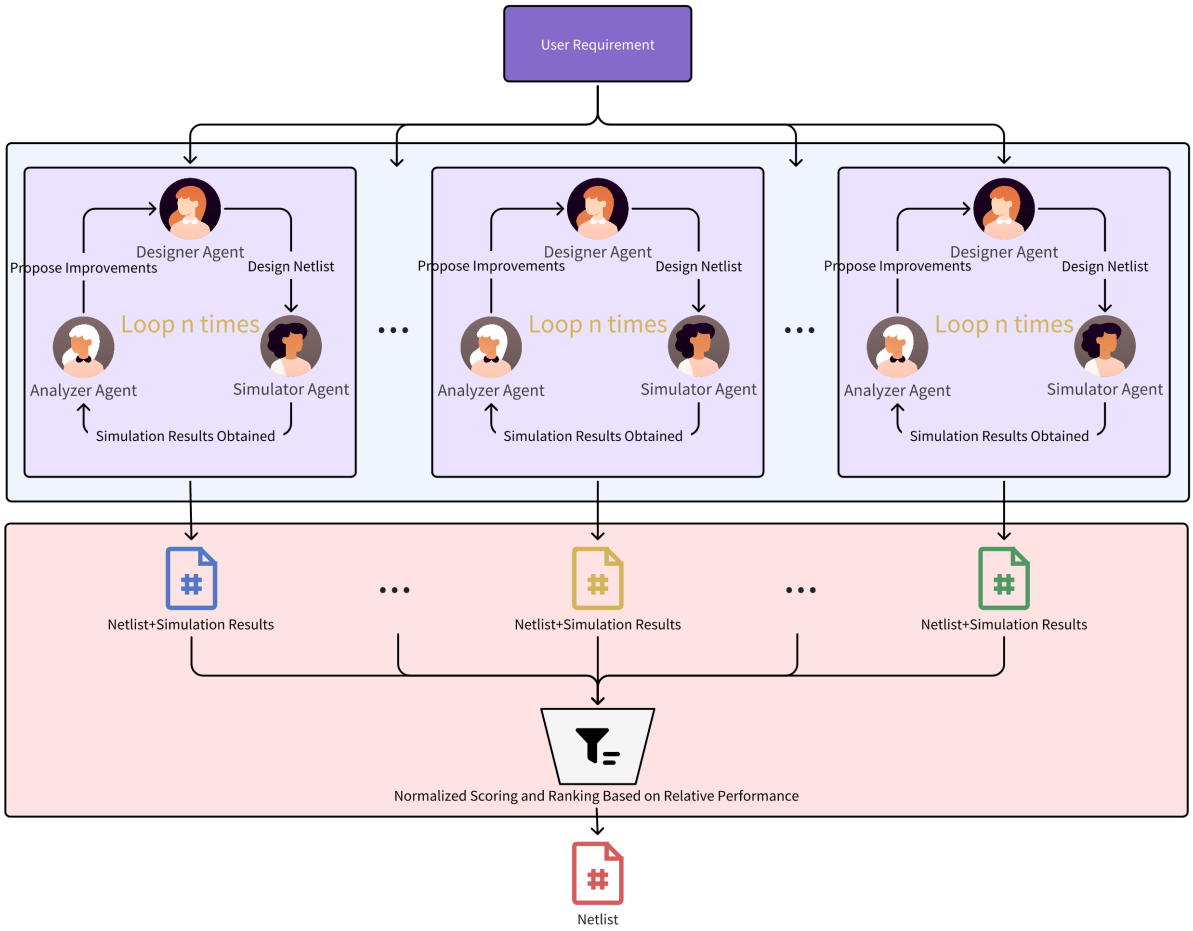

Figure 1: System Architecture and Workflow Diagram

C. Candidate Screening and Optimization

The framework implements batch candidate generation and selection through three coordinated mechanisms, leveraging the LLM's stochastic generation capability combined with quantitative performance evaluation.

1) Parallel Design Generation: The framework generates N independent design trials, where each trial produces a unique circuit variant through temperature-controlled LLM sampling:

$$\mathcal{N}_k \sim p(\cdot|\mathcal{P}, \tau), \quad k = 1, ..., N \tag{7}$$

where τ controls the exploration-exploitation balance via generation diversity. Fault containment is achieved through isolated simulation environments that prevent invalid designs from contaminating the candidate pool.

2) Composite Scoring System: A dual-component evaluation metric combines essential specification fulfillment and relative performance merits:

$$\Psi = \alpha \sum_{j=1}^{5} \mathbb{I}(c_j \geq c_{th}^j) + \sum_{i=1}^{5} beta_i \cdot \frac{s_i - \min_i}{\max_i - \min_i} \tag{8}$$

where:

- $\mathbb{I}(\cdot)$ enforces hard constraints from Table I
- $\min_i, \max_i$ represent bounds of metric s_i
- β_i denotes metric importance weights

3) Adaptive Selection: The optimal design is selected across candidate pools:

$$\mathcal{N}^* = \arg\min_{\mathcal{N}_k} \Psi \tag{9}$$

The top-ranked candidate circuit is returned to the user.

D. Context-Aware Prompt Engineering

The prompt design methodology integrates in-context learning with domain-specific knowledge representation. The template structure follows:

$$\mathcal{P} = [\mathcal{C}_g, \mathcal{C}_d, \mathcal{C}_p, \mathcal{E}] \tag{10}$$

where global constraints $\mathcal{C}_g$ specify topology rules, domain knowledge $\mathcal{C}_d$ contains device models, performance targets $\mathcal{C}_p$ define $A_v > 60dB$, and examples $\mathcal{E}$ demonstrate proper netlist syntax. The curriculum learning strategy gradually introduces complexity levels $l \in \{1, \ldots, L\}$, with constraint subsets $\mathcal{C}^{(l)} \subset \mathcal{C}^{(l+1)}$.

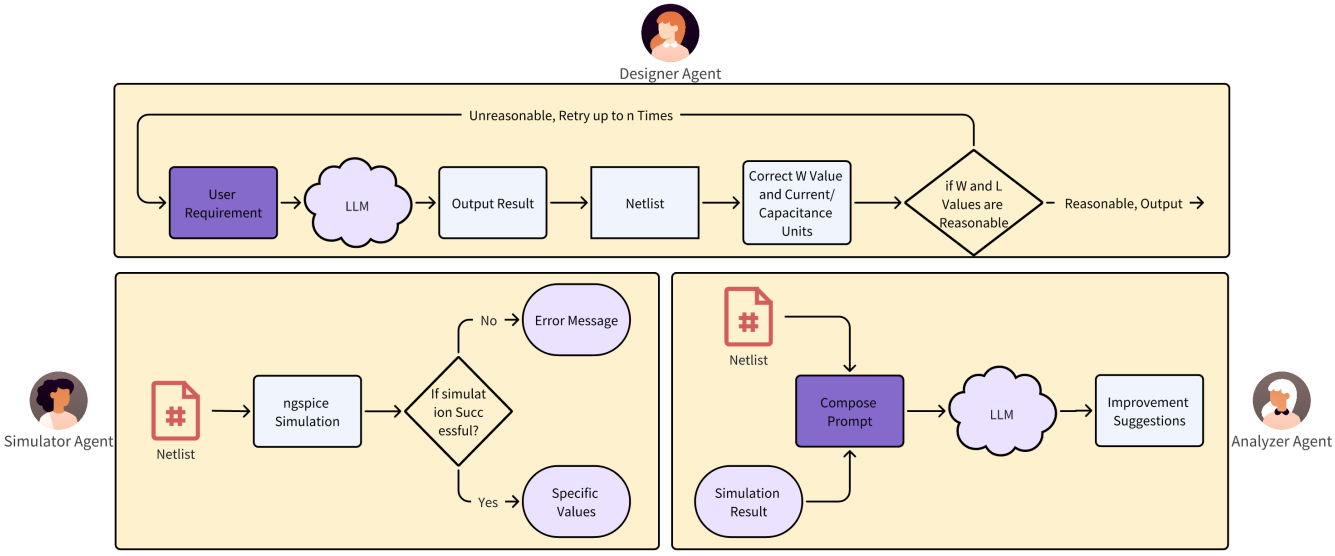

Figure 2: Agent Interaction Protocol and Functional Decomposition

E. Automated Workflow Orchestration

The AmpDesignController implements state machine-based process management with these key features:

Context-Preserving Execution maintains versioned histories $\mathcal{H}_t = \{(\theta_i, \Psi_i, a_i)\}_{i=1}^{t}$ using differential serialization. **Diagnostic Logging** implements tiered verbosity control with error code mapping to resolution strategies.

IV. Experimental Results

The framework is deployed on a server with T-Head Yitian 710 CPU, adopting the open-source pre-trained Qwen2.5-7B as the foundation model. Leveraging the open-source SkyWater SKY130 PDK, the framework autonomously synthesizes transistor-level operational amplifier front-end circuits, with critical performance specifications as follows:

- **Open-loop Gain(Gain)**: 71.0667 dB
- **Gain Bandwidth Product (GBW)**: 35.5660 MHz
- **Phase Margin (PM)**: 62.9380°
- **Slew Rate (SR)**: 20.2350 V/μs
- **Static Current (Idc)**: 454.5440 μA
- **Common-mode Rejection Ratio (CMRR)**: 100.5870
- **Power Supply Rejection Ratio (CMRR)**: 81.4748
- **Equivalent Input Noise Voltage (Noise)**: 0.1186 $\mu V/\sqrt{Hz}$

Experimental results demonstrate that the proposed framework meets the AICAS 2025 Grand Challenge design specifications. Furthermore, the proposed framework currently ranks among the top five of all teams, highlighting its superior performance in terms of both design quality and efficiency. However, it is important to note that due to the inherent uncertainty of large language models, the generated netlists and corresponding performance metrics may exhibit some variability across different runs. This variability is a common characteristic of LLM-based approaches and does not significantly impact the overall effectiveness of the method.

In summary, the experimental results validate the proposed framework's robustness and computational efficiency in automated analog circuit design, capable of consistently synthesizing high quality designs that meet design specifications.

References

[1] S. Zhang, F. Yang, C. Yan, D. Zhou, and X. Zeng, "An efficient batch-constrained bayesian optimization approach for analog circuit synthesis via multiobjective acquisition ensemble," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 1, pp. 1–14, 2022.

[2] W. Lyu, F. Yang, C. Yan, D. Zhou, and X. Zeng, "Batch Bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design," in *2018 35th International Conference on Machine Learning*, pp. 3306–3314, 2018.

[3] W. Cao, J. Gao, T. Ma, R. Ma, M. Benosman, and X. Zhang, "Rose-opt: Robust and efficient analog circuit parameter optimization with knowledge-infused reinforcement learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 44, no. 2, pp. 627–640, 2025.

[4] H. Wang, K. Wang, J. Yang, L. Shen, N. Sun, H.-S. Lee, and S. Han, "Gcn-rl circuit designer: Transferable transistor sizing with graph neural networks and reinforcement learning," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6, 2020.

[5] C. Mattiussi and D. Floreano, "Analog genetic encoding for the evolution of circuits and networks," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 5, pp. 596–607, 2007.

[6] J. Lu, L. Lei, F. Yang, C. Yan, and X. Zeng, "Automated compensation scheme design for operational amplifier via bayesian optimization," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pp. 517–522, 2021.

[7] Z. Zhao and L. Zhang, "Analog integrated circuit topology synthesis with deep reinforcement learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 12, pp. 5138–5151, 2022.