PowerCircuitAI: Agentic Generation of Schematics - The Case of a Three-Level NPC Inverter

Chandan Kumar Sahu†, Matthew Hetrich†, Sneha Narasimhan, Premith Kumar Chilukuri
{*chandan.sahu, matthew.hetrich1, sneha.narasimhan, premith-kumar.chilukuri*}@us.abb.com
ABB US Corporate Research Center

***Abstract*—Power electronic schematic capture remains difficult to automate because a valid converter design must coordinate topology, device ratings, support circuitry, net connectivity. Large language models (LLMs) provide a useful interface for high-level design intent, but unconstrained generation can produce plausible text and code with electrically invalid nets or inconsistent pin assignments. This paper presents a constrained multi-agent workflow for translating natural-language power converter requirements into rule-checked schematic and executable simulations using an agentic pipeline supported by a library of standard topology templates. The proposed methodology selects a canonical circuit graph from the template library, validates graph integrity, and deterministically extracts structural layouts (input connections and phase legs) to generate KiCad schematics and bill-of-materials (BOM). Subsequently, an agentic SPICE generator builds a simulation deck that incorporates self-healing capabilities through automated solver relaxations and agentic repair loops. The method is evaluated on a single-sentence request for a $1.5\,\mathrm{kW}$, $60\,\mathrm{Hz}$, $540\,\mathrm{V}$ dc-bus, three-phase three-level neutral-point-clamped (NPC) inverter. The case study demonstrates that the pipeline successfully resolves the required components, derives the complete detailed schematic, executes SPICE transient simulations outputting the expected three-level pole voltages and multi-level line-to-line staircases, and produces a consistent artifact package. The results show that LLM agents can support the automation of power electronics schematics and simulations for a representative multilevel topology when subject to strict deterministic engineering checks.**

***Index Terms*—Agentic AI, generative artificial intelligence, large language models, power electronics, schematic generation, voltage-source inverter, neutral-point clamped (NPC) inverter, SPICE simulation, electronic design automation.**

I. Introduction

Power electronic converter design is still dominated by expert-driven iteration even though its final artifacts are digital, such as requirements, selected devices, schematics, netlists, simulation, bills of materials, and PCB files. For motor drives and grid-interface converters, the difficulty lies not only in choosing a topology but also in coordinating the continuous-time behavior of high-speed switching cells with device derating, gate-drive constraints, protection, and simulation behavior. Wide-bandgap (WBG) devices intensify this coupling because of high dv/dt and di/dt improve power density while making parasitics and layout geometry more critical [1]. A minor change in trace geometry can introduce commutation-loop inductance that produces destructive transient overvoltage on Gallium-Nitride (GaN) or Silicon-Carbide (SiC) switches [2]. Moreover, physical defects on power boards have caused field failures and product recalls. Consequently, even a conceptually simple three-phase two-level voltage-source inverter (VSI) requires a sequence of structured engineering decisions before the first schematic review, and every decision must remain traceable and verifiable.

A common theme unifies recent advances in generative circuit synthesis: LLMs are stochastic text generators. If they are left unconstrained, they can invent nonexistent components, invalid pinouts, floating nets, contradictory ratings, or syntactically valid code whose electrical meaning is wrong [3]. This stochastic gap is especially hazardous in power electronics, where a missing antiparallel path, a swapped terminal, or an unbounded gate input can invalidate the entire design. This paper argues that LLMs should not be treated as autonomous circuit authorities. Instead, they should be embedded in a *constrained agentic* execution environment in which each generated artifact is checked before it is allowed to influence the next stage. Our prior work *PowerCircuitAI* [4] proposed a fully agentic framework to generate netlists, bill of material, schematics and PCBs. The current work offers a deeper insight into the agentic generation of schematics and validates the schematics with simulation results. The proposed workflow uses specialized agents for topology

synthesis and SPICE deck generation, paired with deterministic algorithms for structural graph analysis and strict verification gates.

The contributions of this work are as follows:

- A generalized multi-agent workflow that converts a natural-language power inverter prompt into machine-readable topologies, KiCad schematics, executable SPICE decks, a bill of materials, and validation output, operating purely on a canonical intermediate representation (CircuitIR) as the single source of truth.
- A deterministic structural graph analysis method that derives component layouts (input and phase legs) from a standard topology template library, ensuring accurate schematic drafting without hallucinating connections and preserving phase symmetry mathematically.
- A SPICE simulation loop that combines deterministic patches, solver-tolerance relaxation, and agentic deck repair to automatically resolve convergence faults and syntax errors without corrupting the underlying connectivity graph.
- A "guardrails-before-autonomy" acceptance model in which every generative act is bounded by observable deterministic checks (graph integrity, structural partitioning, component derating, schematic readability, and SPICE convergence), together with a reproducible case study for a $1.5\,\mathrm{kW}$, $540\,\mathrm{V}$ dc-bus, three-phase three-level NPC inverter detailing complete schematic generation and runtime simulation results.

The remainder of the paper is organized as follows. Section II provides a literature review of generative circuit synthesis. Section III describes the generic constrained agentic framework. Section IV presents the specific three-level NPC inverter case study, including the schematic and simulation results. Section V concludes the paper.

II. Literature Review

A. Generative Circuit Design

Electronic design automation has increasingly explored machine learning and generative modeling for circuit synthesis. Graph and language models have been applied to analog topology generation: *CktGNN* learns circuit-graph representations for design-space exploration [5], *LaMAGIC* maps target specifications directly to circuit graphs via an adjacency-matrix formulation [6], and *AnalogGenie* evolutionarily discovers novel analog topologies [7]. Concurrently, Large Language Models (LLMs) have shifted work toward prompt-driven synthesis: *AnalogCoder* prompts an LLM to emit executable PySpice code [8] and *AmpAgent* sizes multi-stage amplifiers through staged agents [9]. Other works, such as *EEsizer* [10], close the loop toward transistor-level targets, while *Schemato* [11] and *EEschematic* [12] focus on netlist, schematic, and layout translation. However, these methods typically target narrow, pre-defined circuit classes, require task-specific training data, or offload sizing and verification entirely to external simulators. The proposed method overcomes this limitation via a multi-agent workflow operating over standard topology templates with deterministic gates, enabling broad applicability across power-electronic topologies without per-class retraining.

B. Generative AI in Power Electronics

Power converter design incorporates magnetics, thermal physics, and strict spatial constraints, making automated generative design particularly challenging. Recent work includes *PE-GPT*, which couples a multimodal LLM with a knowledge base to design buck converters and dual-active-bridge modulation [13]. Physics-informed agents such as *PHIA* optimize converter modulation [14]. LLM-assisted tools like *Circuit-AI* [15] support bill-of-materials optimization, and programmatic schematic pipelines describe connectivity through domain-specific languages to avoid token-generation errors. Despite these advances, existing tools remain fragmented across sub-tasks, heavily reliant on task-specific Retrieval-Augmented Generation (RAG), and struggle to embed physical rules, leading to hallucination risks [3]. The proposed approach overcomes this fragmentation by utilizing a single CircuitIR source of truth, bounded by strict deterministic electrical guardrails and self-healing SPICE execution, which successfully manages the complexity of demanding multilevel converters.

III. Agentic Framework

The proposed framework converts open-ended natural language requests into a suite of verified engineering artifacts: a canonical circuit graph, KiCad schematics, SPICE netlists and simulation results. The system operates as an *agentic pipeline* that selects a topological skeleton from a curated library of standard power-converter templates. Rather than attempting to derive novel topologies completely from scratch, the agents leverage these known standard templates to ensure

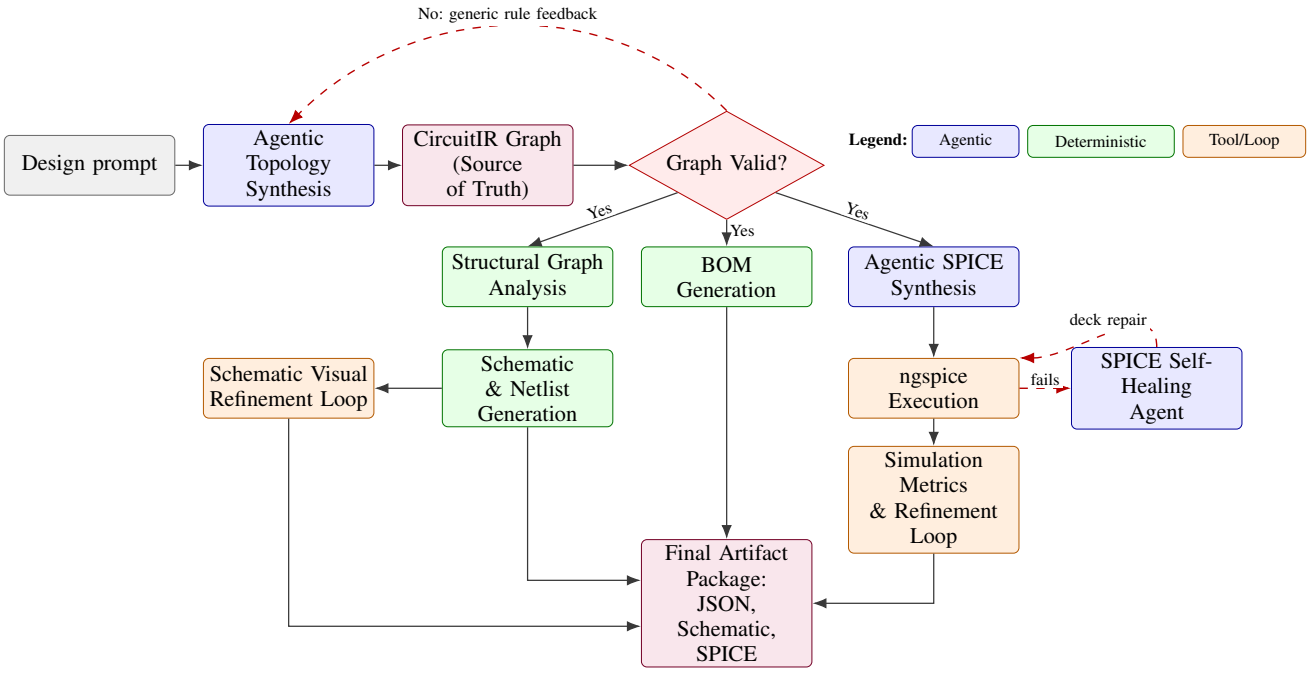


Fig. 1: The constrained agentic workflow for generating power electronic circuits

architectural correctness. The agents are then tasked with resolving devices, instantiating ratings, connecting nets, and executing self-healing simulations dynamically from the prompt. Self-healing refers to the framework's autonomous ability to detect errors during artifact generation or simulation and iteratively apply corrective actions without human intervention. By analyzing deterministic error logs and adjusting parameters, the system recovers from faults to produce a viable final design. In this framework, the term *agent* describes an LLM-backed actor that reasons over a bounded subtask with typed inputs and outputs. Table I contrasts the roles of LLM agents with deterministic stages. The system relies on the principle that autonomy is only safe when bounded by strict engineering guardrails. Each generative act is subject to a deterministic check, ensuring that artifacts are accepted based on observable rules (e.g., fan-out counts, ngspice convergence, and layout metrics) rather than on the opaque confidence of a language model. This strategy guarantees that the system produces robust, reproducible circuit designs across general specifications without hallucinating non-viable structures.

Fig. 1 summarizes the computational workflow. The pipeline is architected as a stack of subsystems, in which each layer operates solely on a canonical Intermediate Representation (CircuitIR). This strict separation isolates failures so that a localized defect degrades to a warning rather than aborting the entire output package.

A. Agentic Topology Synthesis

The entry point to the workflow is the Topology Synthesizer agent. Given a natural-language request (e.g., "A three-level NPC inverter for a 540V DC bus"), the agent selects the appropriate base template and generates a comprehensive `CircuitIR`. This canonical data model specifies the components, their generic roles (e.g., *freewheel*, *series_switch*), named nets, connectivity mappings, and design objectives. The generated `CircuitIR` undergoes topology-agnostic electrical validation to catch floating pins, duplicate designators, and missing grounds. If the graph validation fails, the validation errors are fed back into the agent to close an automatic repair loop.

B. Structural Graph Analysis and Deterministic Schematic Generation

To layout the schematic efficiently, the system performs a deterministic structural analysis on the validated bipartite component-net graph guided by the selected template. The structural analyzer uses a fan-out algorithm to identify horizontal busbars (rails): it ranks nets by their connection degree and selects the highly connected power and neutral lines. Components that sit between these identified rails are then aggregated into vertical series-leg groupings. A breadth-first search

TABLE I: Agents and the deterministic stages

Stage	LLM / Agent Role	Deterministic / Tool Role	Output Artifact	Acceptance Basis
Topology Synthesis	Parses prompt and selects template	Validates electrical graph integrity	Canonical CircuitIR JSON	Required topology nodes present; passes electrical checks
Structural Analysis	Normalizes free-text role hints	Deterministically detects rails and series legs	Structural layout model	Derived structure maps to connected components
Schematic Generation	Iterative readability critique (Visual Inspector)	Composes KiCad schematic utilizing generic layout backend primitives	`.kicad_sch` and netlist	Deterministic readability scores (overlapping symbols, dangling wires)
BOM Extraction	Optional enrichment and candidate review	Retrieves exact MPN	BOM CSV/JSON	Rated components exceed required margins and are procurable
SPICE Execution	Synthesizes live simulation deck and self-heals faults	Injects structure-triggered playbooks and runs `ngspice` engine	Executed `.cir` deck and waveform plot	Deck successfully converges and waveforms satisfy sanity checks

traversing from the positive rail downwards computes a top-to-bottom ordering for the devices within each leg.

These derived rails and legs become the direct inputs for the schematic generator, which translates the generic columns into a `.kicad_sch` document. Because structurally identical legs hash equally, phase symmetry is preserved mathematically. The resulting layout is then reviewed by a visual agent and corrected through a schematic refinement loop targeting symbol overlap and visual crowding.

C. Deterministic Acceptance Criteria

The workflow deliberately avoids relying on LLM confidence as a correctness metric. Table II details the deterministic constraints imposed by the execution environment. Failures at the graph or SPICE validation levels typically trigger automated agentic repair, while failures in structural analysis or schematic integrity fall back to known-good flat layouts.

D. SPICE Deck Generation and Self-Healing Execution

Simulation decks are constructed agentically but benefit from deterministic hygiene. The framework analyzes the `CircuitIR` to simulate a switching leg if one is detected. The generated deck is then executed via an embedded `ngspice` solver. A three-tier self-healing ladder addresses simulation failures: first applying deterministic patches, then relaxing solver tolerances, and finally invoking a specialized *SPICE Fixer* agent. The agent examines the error text alongside the `CircuitIR` to perform surgical repairs on the netlist without ever corrupting the underlying connectivity graph.

E. Agentic Refinement and Artifact Export

The pipeline concludes with a generic refinement engine that iterates over generated schematics and simulation waveforms. By rendering an artifact to an observable form (such as an SVG schematic or waveform plot) and scoring it deterministically, a vision agent can critique its legibility and apply bounded repairs. The iterative agentic refinement stops upon saturation of the metrics. An overarching artifact exporter orchestrates the creation of netlists, a bill of materials (BOM), schematic, simulations and other intermediate artifacts, packaging all live-generated engineering data for final human review.

IV. Case Study: DC-to-AC Three-Phase Three-Level NPC Inverter

The framework is evaluated using the generic prompt: *"DC-to-AC three-phase three level NPC topology for a 1.5 kW, 60 Hz load with a 540 V DC bus."* The prompt is intentionally compact. It states the converter function, the neutral-point-clamped (NPC) topology, power level, fundamental frequency, and dc-bus voltage. The prompt does not prescribe the semiconductor part, the split dc-link arrangement, the number of devices per leg, the clamp-diode placement, the report format, or the schematic net naming. Relative to a two-level bridge, the three-level NPC roughly doubles the device count and introduces a floating neutral point, making it a demanding test of whether the workflow can synthesize a topologically correct, traceable, and generic set of engineering artifacts.

A. Specification and Electrical Assumptions

Table III lists the structured specification derived by the pipeline. The topology synthesizer generated a `CircuitIR` graph representing a three-level NPC inverter with three phase legs, a split dc link, and a clamped neutral point based on the selected standard topology template. Each leg comprises four series switches (an outer and inner device on both the high and low side), four antiparallel diodes, and two neutral-point clamping diodes, yielding 12 active switch positions

TABLE II: Acceptance criteria used to prevent electrically inconsistent artifacts from propagating.

Check	Failure Mode Caught	Pass Criterion	Feedback Behavior
Graph Integrity	Floating pins, unknown nets, missing system ground	Topology graph resolves cleanly without basic electrical rule violations	Feeds generic error logs back to the synthesis agent for repair
Structural Leg Analysis	Highly tangled fan-out graphs causing ambiguous leg groupings	Graph correctly partitions into identifiable horizontal rails and series legs	Restructures groupings or falls back to flat generic placements
Component Derating	Underrated semiconductors or incorrect voltage/current assignments	Selected component ratings explicitly exceed application requirements	Limits BOM candidates during supplier lookup and alerts refinement loop
Schematic Integrity	Empty hierarchical roots, dangling netlist endpoints, or symbol collisions	Schematic file is self-contained and scores sufficiently on readability metrics	Triggers agentic visual fixes or reverts to a guaranteed flat page fallback
SPICE Convergence	Invalid subcircuit syntax, missing node declarations, or simulator blowup (NaN)	Deck is generated, passes `ngspice` parsing, and completes transient solve	Iterates self-healing solver relaxations and agentic deck repair

TABLE III: Structured specification inferred from the natural-language prompt.

Parameter	Resolved value
Topology Synthesized	Three-level, three-phase NPC
Rated output power	1.5 kW
Nominal dc-bus voltage	540 V
DC-link arrangement	Split, neutral point clamped
Estimated rated line current	2.91 A
Output frequency	60 Hz
Switching frequency	10 kHz
Split dc-link capacitance	2×680 µF
Component / net count	35 / 24

and 6 clamping diodes across the three phases. The estimated line-line fundamental voltage is 297.4 V rms for the selected modulation assumptions, and the rated line current is 2.91 A. The reported current is based on

$$I_{\text{line}} = \frac{P_{\text{out}}}{\sqrt{3} V_{LL,1} \eta \cdot \text{PF}}, \tag{1}$$

where $P_{\text{out}} = 1.5\,\text{kW}$, $V_{LL,1} = 297.4\,\text{V}$ is the estimated fundamental line-line rms voltage, and the first-pass sizing assumes $\eta \cdot \text{PF} \approx 0.98$. All RMS quantities in this section refer to fundamental line-line quantities.

The agentic synthesis mapped the components to 12 active semiconductor positions, 6 neutral-point clamping diodes, and a split dc-link block of two series capacitors defining the positive rail (`DC_BUS_P`), negative rail (`DC_BUS_N`), and midpoint (`DC_BUS_NP`). Each phase exposes two internal nodes (`NODE_x1`, `NODE_x2`) between the series switches. The structural graph analyzer then deterministically verified the three-phase legs and the shared neutral point directly from the bipartite component-net connectivity, confirming that the clamping diodes of every leg terminate on the common midpoint.

B. Component Selection and Derating Evidence

Table IV summarizes the quantitative selection evidence reported by the BOM extraction stage. Because the NPC structure halves the voltage stress on each series device to one dc-link half ($\approx$ 270 V), a 600 V-class device provides adequate margin. The resolved candidate is an 600 V-class IGBT (`IKW75N60H3`). While this provides the necessary voltage blocking capability, the 75 A nominal current rating significantly oversizes the device relative to the 2.91 A line current requirement. The part is instantiated 12 times to populate the four switch positions in each of the three legs, complemented by 6 neutral-point clamping diodes.

TABLE IV: First-pass component-selection and derating evidence.

Item	Requirement / assumption	Resolved value
Switch positions	Four per phase leg	12 active switches
Clamping diodes	Two per phase leg	6 clamp diodes
Per-device blocking	Half dc link ≈ 270 V	600 V class
Device candidate	IGBT class	IKW75N60H3
Line current basis	Eq. (1)	2.91 A
Switching frequency	Prompt/default sizing	10 kHz

C. Generated Artifacts and Schematic Details

The output includes both human-readable and machine-readable records: a canonical `CircuitIR` JSON, the composed topology, a KiCad schematic (`.kicad_sch`), an executable SPICE netlist (`.cir`), the complete simulation results, and the closed-loop verification records. The pipeline completed across topology synthesis, structural analysis, schematic generation, and SPICE self-healing with zero fatal errors, terminating in a *completed-with-warnings* state.

The generated schematic explicitly resolves the three NPC phase legs, connecting the positive and negative dc rails and the clamped neutral point to the output nodes (U, V, W). A deterministic NPC electrical-rule check verified switch–antiparallel-diode pairing across all 12 switch positions in the three phases, reporting zero errors and zero warnings. The visual inspection agent

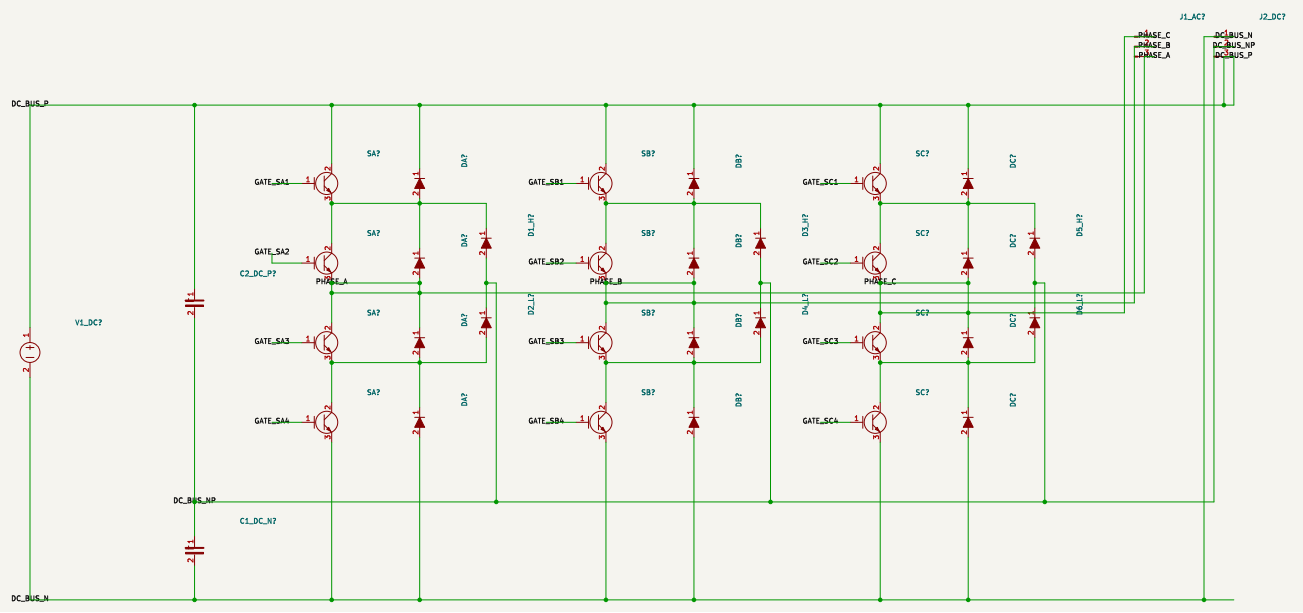


Fig. 2: Generated KiCad schematic for the *"DC-to-AC three-phase three level NPC topology for a 1.5 kW, 60 Hz load with a 540 V DC bus."*

scored the layout at 8.39 on the aggregate readability metric (a heuristic weighted sum evaluating symbol collisions, wire crossings, and component spacing), with a unity symbol-overlap score confirming the absence of overlapping symbols.

D. SPICE Simulations and Self-Healing Execution

Beyond producing connectivity, the agentic framework generated an `ngspice`-compatible simulation deck to support transient functional testing. The agent synthesized a behavioral deck for the standard split dc link topology, modeling the upper and lower 270 V halves with their 680 µF capacitors and shared midpoint, together with voltage-controlled switch and diode models, level-shifted SPWM comparator gate sources with a dead-time margin and a balanced three-phase series R–L load referenced to a common star point.

Crucially, the system executed the SPICE simulation dynamically, resolving convergence and syntax anomalies through the agentic self-healing loop and injecting deterministic rule checks. The transient solve converged and produced the expected three-level pole behavior across 36 257 sample points over a 30 ms duration: each phase node steps among the approximately 0 V, 270 V, and 540 V levels referenced to the negative rail. Specifically, the simulation confirmed RMS phase voltages of approximately 310 V–355 V and fundamental line-line RMS voltages clustering around 286 V–316 V, forming the characteristic multilevel staircase across the balanced 10 Ω + 10 mH three-phase load and driving continuous phase currents of 14.1 A–15.7 A RMS. Table V details the verification output from the SPICE runtime environment, and Fig. 3 illustrates the complete multilevel switching transient.

TABLE V: Verification status and closed-loop acceptance for the generated artifact set.

Metric	Status / value
Pipeline status	Completed with policy warnings
Circuit graph validation	Pass (no duplicate refs/unknown nets)
NPC switch–diode ERC	Pass (12 pairs, phases A/B/C)
KiCad schematic generated	Yes (structural analysis layout)
Schematic readability score	8.39 / 10 aggregate
SPICE static validation	Pass (accepted netlist)
Self-healing SPICE convergence	Pass (executed ngspice)
Three-level pole behavior	Verified (≈0/270/540 V)

The generic framework demonstrated that an LLM can provide correct three-level topological connectivity via `CircuitIR` based on standard topology templates, while deterministic structure extraction and the NPC pairing ERC prevent drafting hallucinations. The SPICE self-healing refiner iterated to convergence independently, yielding executable waveform data that exhibits the multilevel switching signature.

E. Discussion and Limitations

The case study illustrates that generic, constrained agentic design can bridge the gap from a natural-language description of a non-trivial multilevel topology directly to logically consistent schematic artifacts and a working SPICE simulation. The results establish that logical correctness and connectivity, including the split dc link, floating neutral point, and clamp-diode network

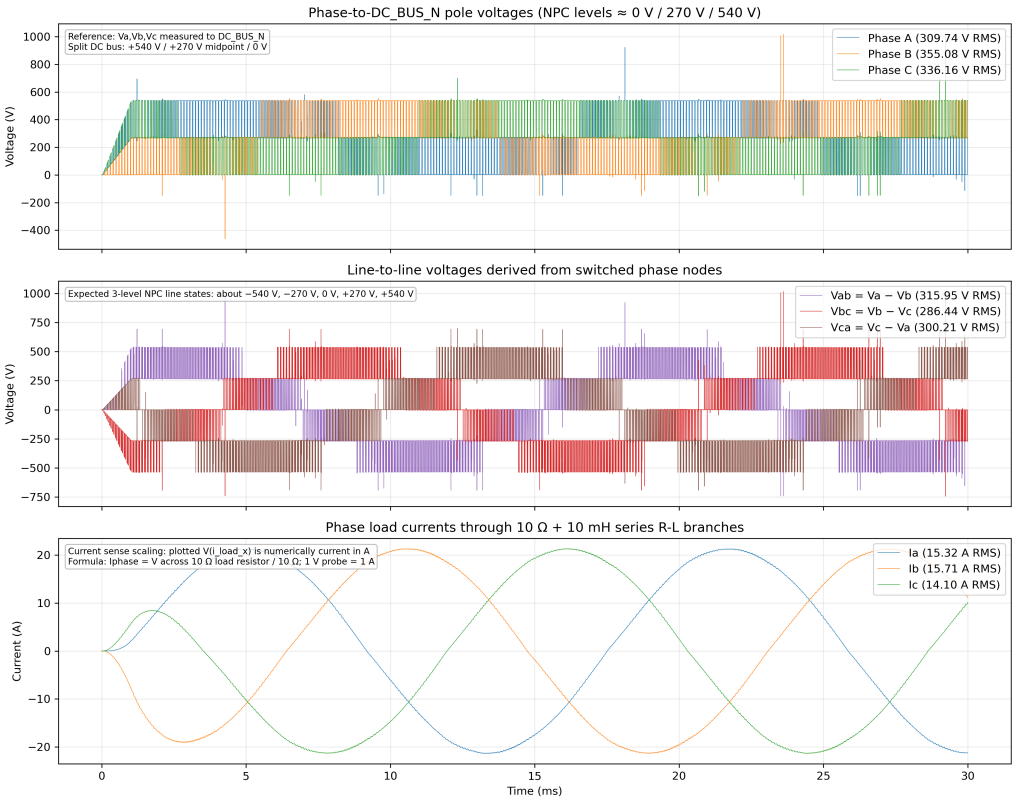


Fig. 3: Self-healed SPICE simulation results for the generated three-level NPC inverter. Top: Pole voltages measured against the negative DC rail, showing three discrete switching states (≈0/270/540 V). Middle: Derived line-to-line voltages exhibiting the expected five-level staircase pattern. Bottom: Filtered sinusoidal load currents through the R-L branches (10 Ω, 10 mH) with RMS values correctly balanced across phases.

unique to the NPC family, can be automated when bounded by strict programmatic guardrails.

Several limitations remain, and the framework surfaces them honestly rather than concealing them. First, the run resolved under a conceptual-mode diagram policy that deliberately suppressed inferred default add-ons (precharge) and auxiliary integrity injection. Consequently, the cross-artifact consistency checker flagged that the design report hallucinates parameters for precharge, EMI filtering, and protection subsystems that are absent from the schematic and SPICE artifacts. These warnings highlight a critical limitation of isolated agentic tasks (agentic myopia): while a topology agent may correctly constrain itself to the core NPC bridge, a downstream report-writing agent may infer system-level requirements that were never built. While the framework's guardrails successfully caught this scope creep, the hallucination itself remains a failure mode of the reporting stage, emphasizing the need for tighter cross-agent state alignment and higher-level architectural orchestration in future multi-agent systems.

Second, the simulation uses ideal voltage-controlled switches and simplified diode models rather than vendor electrothermal models. Large language models struggle to accurately parameterize complex, non-linear vendor SPICE subcircuits without extensive use of retrieval-augmented generation (RAG) and tools. Therefore, the generated deck validates the switching logic and three-level pole formation but cannot accurately predict device losses or precise RMS magnitudes. Integrating real-world electrothermal physics remains a significant hurdle for pure LLM generation.

Third, while LLMs excel at the symbolic reasoning required for 2D schematic connectivity, they lack the spatial and physics-based intuition required for high-power physical design. NPC-specific concerns, such as neutral-point voltage balancing and unequal clamp-diode loss distribution, along with layout challenges like minimizing commutation-loop inductance and adhering to creepage/clearance rules, require 3D physics-aware placement. Furthermore, the general problem of optimal component placement and routing is known to be NP-hard [16], placing robust physical layout generation fundamentally outside the capabilities of pure stochastic text generation. The generic framework intentionally isolates the schematic-drafting success from physical

manufacturing, establishing a clear boundary between semantic connectivity and physical routing.

Finally, while the pipeline incorporates deterministic checks, the underlying LLM generation remains stochastic. The results presented reflect a single representative execution. Future work must rigorously quantify the statistical reliability and iteration overhead of the synthesis and self-healing loops across a broader corpus of converter topologies.

V. Conclusion and Future Work

This paper presented a constrained multi-agent workflow for the agentic generation of power-electronic schematics and executable simulations. The central result is that LLM agents can effectively support converter design when their generative capabilities are integrated into an agentic pipeline operating on a curated library of standard topology templates, subject to strict deterministic engineering checks. The proposed framework decomposes the task into agentic topology synthesis from standard templates, deterministic structural graph analysis for automated layout, KiCad schematic generation, BOM enrichment, and self-healing SPICE simulation execution.

The methodology's architecture was demonstrated through a specific case study: a $1.5\,\mathrm{kW}$, $60\,\mathrm{Hz}$, $540\,\mathrm{V}$ dc-bus, three-phase three-level neutral-point-clamped (NPC) inverter generated from a single-sentence prompt. The pipeline successfully produced a detailed, rule-checked artifact set, including a canonical CircuitIR graph, a KiCad schematic, an executable SPICE deck, and simulation waveforms. The system accurately identified and laid out the expected twelve active switch positions, six clamping diodes, a split dc-link, and shared neutral point based on the selected standard topology template, executing the simulations and self-healing without requiring human intervention. The results successfully reproduced the target $270\,\mathrm{V}/540\,\mathrm{V}$ split-bus behavior and expected current profiles in a complete transient simulation.

This work successfully establishes a robust foundation for automated, prompt-driven power schematic design and simulation. While the framework delivers fully detailed schematics and transient simulations for generic power converters using standard templates, future work will extend the methodology to integrate physics-aware placement and routing, thermal management, and 3D electromagnetic compliance checks to bridge the gap toward fully manufacturable hardware boards.

References

[1] A. K. Morya *et al.*, "Wide bandgap devices in ac electric drives: Opportunities and challenges," *IEEE Transactions on Transportation Electrification*, vol. 5, no. 1, pp. 3–20, 2019.

[2] V. Baker *et al.*, "3d commutation-loop design methodology for a silicon-carbide based 15 kw, 380: 480 v matrix converter with pcb aluminum nitride cooling inlay," in *2020 IEEE energy conversion congress and exposition (ECCE)*. IEEE, 2020, pp. 233–238.

[3] Z. Ji *et al.*, "Survey of hallucination in natural language generation," *ACM computing surveys*, vol. 55, no. 12, pp. 1–38, 2023.

[4] M. Hetrich *et al.*, "Power circuit ai: Designing power electronic circuits for motor drives with generative artificial intelligence," in *2026 IEEE International Conference on Intelligent Design and Control of Automation and Drive Systems (IDCD)*. IEEE, 2026, pp. 1–7.

[5] Z. Dong *et al.*, "Cktgnn: Circuit graph neural network for electronic design automation," *arXiv preprint arXiv:2308.16406*, 2023.

[6] C.-C. Chang *et al.*, "Lamagic: Language-model-based topology generation for analog integrated circuits," *arXiv preprint arXiv:2407.18269*, 2024.

[7] J. Gao *et al.*, "Analoggenie: A generative engine for automatic discovery of analog circuit topologies," *arXiv preprint arXiv:2503.00205*, 2025.

[8] Y. Lai *et al.*, "Analogcoder: Analog circuit design via training-free code generation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 1, 2025, pp. 379–387.

[9] C. Liu *et al.*, "Ampagent: An llm-based multi-agent system for multi-stage amplifier schematic design from literature for process and performance porting," *arXiv preprint arXiv:2409.14739*, 2024.

[10] C. Liu and D. Chitnis, "Eesizer: Llm-based ai agent for sizing of analog and mixed signal circuit," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2025.

[11] R. Matsuo *et al.*, "Schemato–an llm for netlist-to-schematic conversion," in *2025 ACM/IEEE 7th Symposium on Machine Learning for CAD (MLCAD)*. IEEE, 2025, pp. 1–7.

[12] C. Liu and D. Chitnis, "Eeschematic: Multimodal-llm based ai agent for schematic generation of analog circuit," in *2026 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2026, pp. 822–826.

[13] F. Lin *et al.*, "Pe-gpt: A new paradigm for power electronics design," pp. 3778–3791, 2024.

[14] J. Liu *et al.*, "Physics-informed autonomous llm agents for explainable power electronics modulation design," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 40, no. 47, 2026, pp. 40 063–40 072.

[15] V. Raval, M. Zeid, and P. Enjeti, "Circuit-ai: An advanced large language model (llm) based ai-agent for bill of materials (bom) optimization, circuit simulations & design," in *2025 IEEE Energy Conversion Conference Congress and Exposition (ECCE)*. IEEE, 2025, pp. 1–7.

[16] M. R. Garey and D. S. Johnson, "The rectilinear steiner tree problem is np-complete," *SIAM Journal on Applied Mathematics*, vol. 32, no. 4, pp. 826–834, 1977.