

From Ad Hoc Simulations to Orchestrated RF Design Workflows

Executive Summary

With RF circuit and system design becoming more complex, tighter performance targets, more design variables and growing pressure to shorten engineering cycles means that traditional manual simulation workflows are difficult to repeat consistently and do not scale well across teams or projects. Python-based automation improves repeatability and efficiency, but many RF engineers are not full-time programmers, which can make code-only workflows difficult to adopt and maintain.

A flowchart-based orchestration model offers a practical alternative. By combining executable visual workflows with Python-backed action blocks, engineering teams can capture simulation intent, decision logic and optimization steps in a format that is easy to understand, reuse and automate. This whitepaper explains how this approach can support RF power amplifier design, using an intrinsic-node waveform engineering example to show how simulation tasks, algorithms and decisions can be integrated into a single repeatable workflow.

- **Challenge:** Manual and ad hoc RF simulation workflows are difficult to scale, document and reproduce.
- **Approach:** Executable flowcharts combine visual process capture with Python automation for repeatable RF simulation orchestration.
- **Example:** A Class J RF power amplifier workflow demonstrates DCIV analysis, load-line calculation, harmonic tuning, source-pull, S-parameter simulation, harmonic balance and optimization.

Outcome: Engineering teams can improve traceability, accelerate iteration, and create clearer interfaces for AI-enabled design assistance.

The Challenge: Scaling RF Simulation Workflows

RF simulators have evolved from command-line tools to graphical user interfaces that simplify setup for individual analyses. GUIs are intuitive and accessible, but they can introduce workflow variability. Achieving consistent results across repeated simulation runs requires the same inputs, settings, and sequence of steps every time. Reusable templates can help, but they often remain editable, making process traceability difficult.

Scripting improves repeatability by enabling automation, parameterization, and reuse. Python has become especially important because of its broad open-source ecosystem, portability, and suitability for simulation automation. However, script-centric workflows can create barriers for engineers who have deep RF design expertise but limited interest in maintaining complex codebases.

The design challenge is therefore not simply whether to use GUIs or code. Engineering organizations need a workflow model that preserves the accessibility of visual tools while providing repeatability, scalability and decision logic of automation.

The Solution: Executable Flowchart-Based Orchestration

Keysight's RF Circuit Simulation Professional platform, Nexus, is designed with automation and AI-readiness in mind. Building on Nexus, Nexus Connect enables RF engineering teams to capture design processes as executable flowcharts. Each workflow is composed of action blocks, or steps, that are underpinned by modifiable Python code. The result is a no-code, low-code or fully coded orchestration environment that can adapt to the needs and preferences of different users.

This approach allows engineering logic to be documented and executed in the same environment. Simulation setup, parameter selection, optimization goals and decision criteria are captured in a reusable flow rather than scattered across notes, scripts or one-off GUI actions.

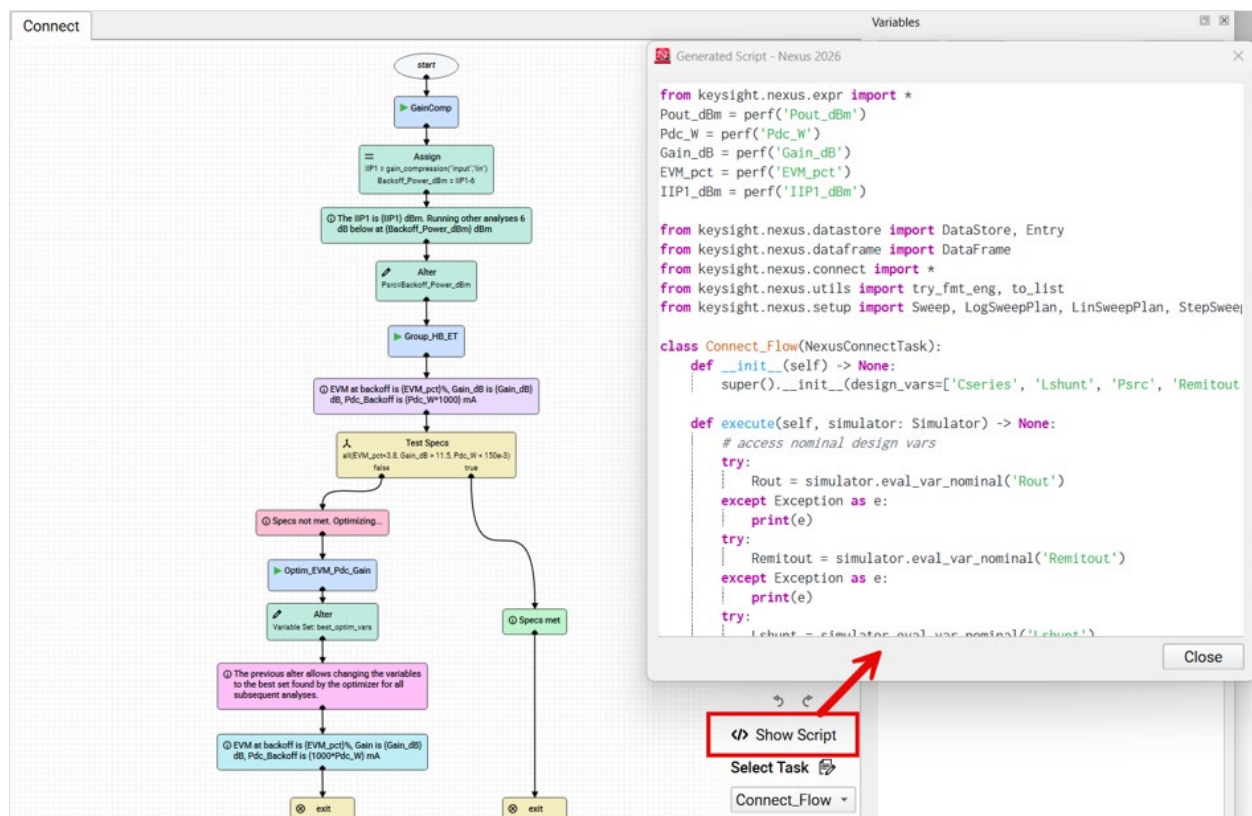


Figure 1. A Nexus Connect flowchart and its associated Python code.

Key Capabilities

- Capture RF simulation workflows as visual, executable diagrams.
- Reuse validated workflows across design projects and teams.
- Embed Python logic where deeper customization or automation is required.

Make simulation decisions, optimization criteria, and process assumptions transparent.

- Create structured interfaces that future AI agents and skills can invoke.

Visual workflow diagrams help engineering teams capture and share design intent from the start of a project. Instead of relying on physical whiteboards, disconnected scripts or informal knowledge transfer, teams can see which simulations were selected, which parameters were used, which optimization goals were applied, and what criteria determined the next step. The flowchart becomes both an executable process and a communication tool for design reviews, onboarding and reuse.

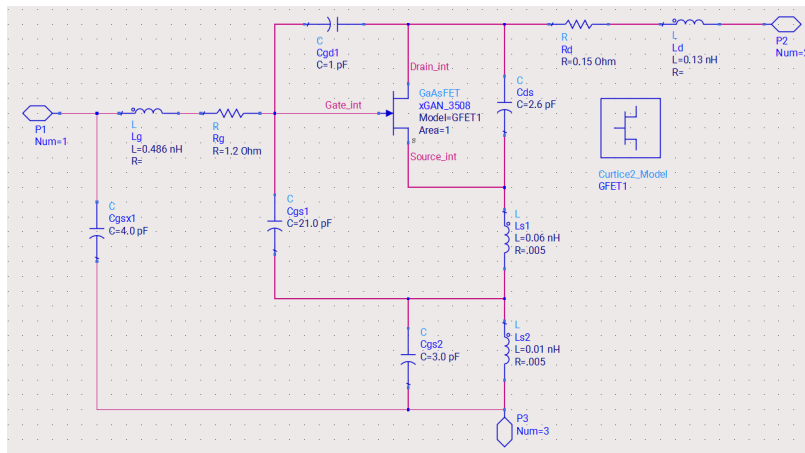


Figure 2. An intrinsic-node GaN transistor model with a current generator and parasitics, represented in ADS.

Technical Case Study: Class J Power Amplifier Design

To demonstrate the value of flowchart-based orchestration, this whitepaper uses an RF power amplifier design example based on a Class J amplifier. Class J operation is similar to Class B, but it uses harmonic tuning to improve efficiency. The example is well suited to workflow orchestration because PA design requires multiple interdependent simulation tasks and iterative decisions.

The workflow uses Advanced Design System (ADS) for schematic capture and data display, and Nexus for simulation execution. The design method focuses on intrinsic-node waveform

engineering, which analyzes voltages and currents inside the device rather than only at external terminals.

In contrast, intrinsic design methodologies analyze voltages and currents at internal device nodes rather than at terminals. By working closer to the intrinsic current generator, designers can directly observe waveforms that are typically masked by device parasitics. This perspective provides deeper insight into transistor operation and enables the evaluation of key PA performance metrics. Figure 2 shows a schematic of a conceptual, non-process-specific GaN transistor with its intrinsic device nodes explicitly exposed.

Why Intrinsic-Node Analysis Matters

Intrinsic-node access allows PA designers to build around first principles. By observing waveforms closer to the intrinsic current generator, engineers can evaluate transistor behavior that would otherwise be obscured by parasitic resistance, capacitance and inductance. The challenge is that an intrinsic-node approach can involve many manual steps unless the process is orchestrated and automated.

- Ideal nonlinear voltage and current waveforms are superimposed on a device's DCIV curves to determine the power and efficiency a device could achieve in that operating state.
- Strategically engineered waveforms reduce overlap between the time-domain voltage and current waveforms, minimizing resistive power dissipation.
- The Class J mode tunes the second-harmonic load impedance to boost the voltage waveform, while higher harmonics shape the current waveform, theoretically allowing an efficiency of 78.5 percent with a more flexible set of harmonic impedances than in Class B.
- The time-domain waveforms are then transformed into the frequency domain using an FFT, yielding computed impedances at the fundamental and harmonic frequencies.
- By applying those impedances back to the idealized current generator in the intrinsic-node transistor model, energy organizes into the efficient waveforms we started with, maximizing achievable device power efficiency.

The PA design bench schematic in ADS, as seen in Figure 3, mimics a load-pull measurement system. It includes harmonic tuners, bias networks, and receivers for the A (incident) and B (reflected) waves, enabling a complete analysis of PA metrics. The device under test (DUT) is the GaN FET model shown in Figure 2.

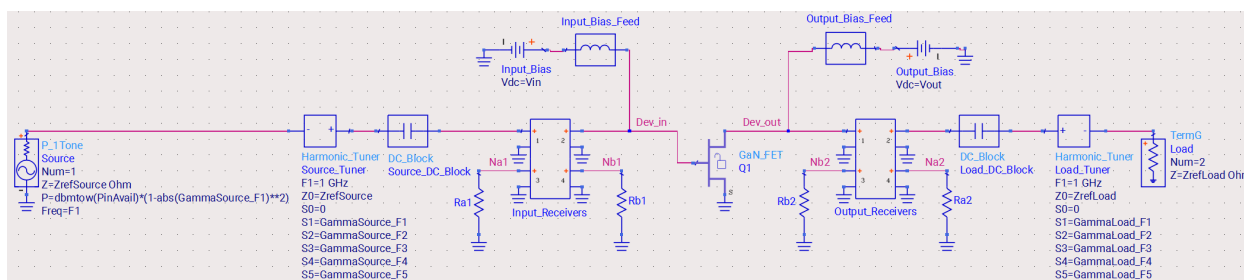


Figure 3. PA design bench schematic in ADS using the GaN FET model.

Orchestrated Workflow

A single executable Nexus Connect task captures the sequence of analyses and decisions needed to complete a significant portion of the Class J amplifier design. The workflow integrates simulation setup, algorithmic calculations, optimization and final performance confirmation.

- First, the task performs a DCIV sweep, followed by load-line calculations for bias and target intrinsic impedances (Z_s) at each harmonic.
- Then, it sequentially optimizes extrinsic loads for each harmonic to achieve the target intrinsic Z_s .
- Next, it performs a source-pull analysis and an input power adjustment to match power and efficiency.
- Then, an S-parameter analysis is performed, including stability calculations with Winslow virtual probes.

Finally, a power-swept HB analysis confirms the overall PA performance.

As the flowchart execution graph progresses, run blocks can change color according to a programmable setting. Double-clicking a block displays the associated simulation or optimization results. Figure 4 also presents a portion of the results from the first load-pull optimization analysis, which appears when the graph is clicked. Designers can define top-level parameters, including the target output power; alpha, which reshapes the voltage waveform and alters the target impedances without affecting power or efficiency; and the quiescent (q)-point current as a percentage of the device's maximum output current. When the flowchart is rerun with revised top-level parameters, updated results are generated quickly—in less than a minute on a laptop.

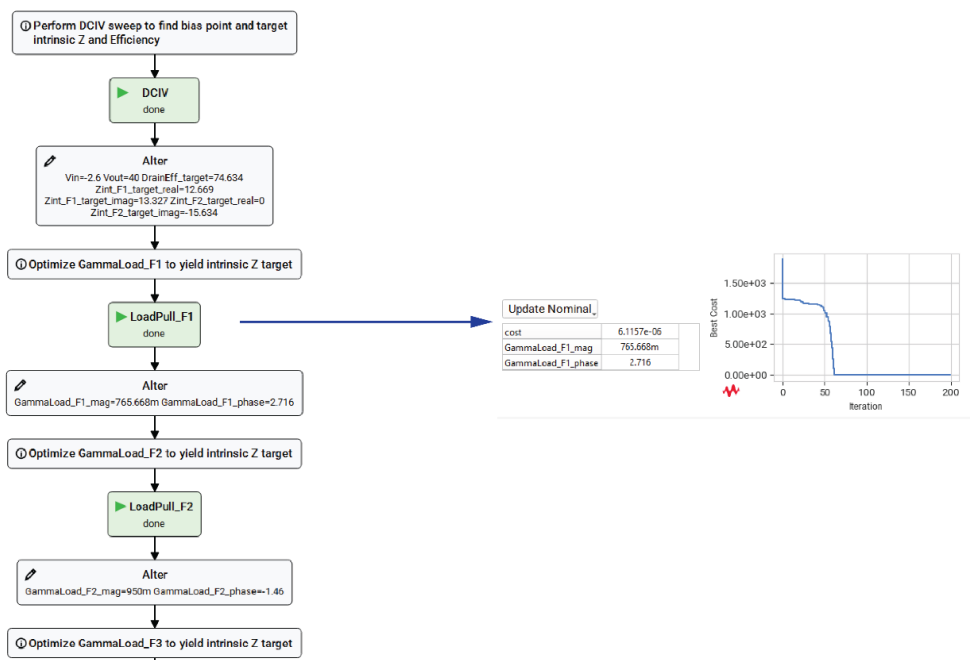


Figure 4. A Nexus Connect task execution flowchart with results of a load-pull optimization displayed.

Deriving target intrinsic gammas involves constructing a load line from the DCIV sweep data by connecting the q point to one of the knee points. First, simple Python algorithms and heuristics determine the q point from the device's breakdown voltage and the desired output current at cutoff (a small percentage of the maximum output current). Then, knee points are calculated for all simulated V_{in} curves, as shown in Figure 5.

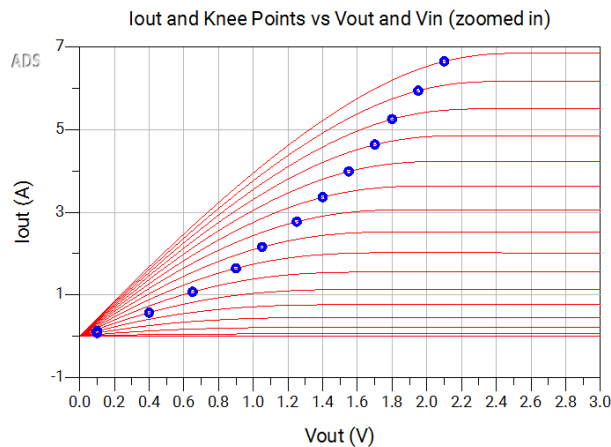


Figure 5. DCIV sweep data plotted with all the calculated knee points.

Next, load-line metrics are evaluated for all knee/ quiescent point pairs, and additional Python algorithms are used to select a load line from the candidates. The algorithms balance output power and efficiency while walking down the knees. (The actual details of our algorithms are not the focus here — the key point is that all decisions and rules are formally captured in the execution flowchart.) The blue line in Figure 6 is the rectified load line (or IV swing at α equals 0.0). With the FET biased for Class J operation at the q point, the instantaneous current and voltage swing along the line when an RF signal is applied. The FET conducts zero current for half of the RF cycle, yielding a 50 percent duty cycle. With an α of 0.0, the voltage waveform is a pure sinusoid.

Using α values of 1.0 or -1.0 gives a "harmonic boost" to the voltage waveform, moves the target intrinsic gammas off the real axis of the Smith chart and widens the IV swing trajectory away from the rectified load line shown in teal.

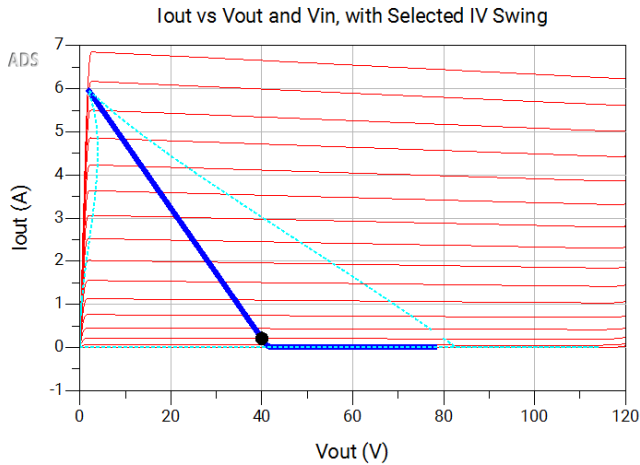
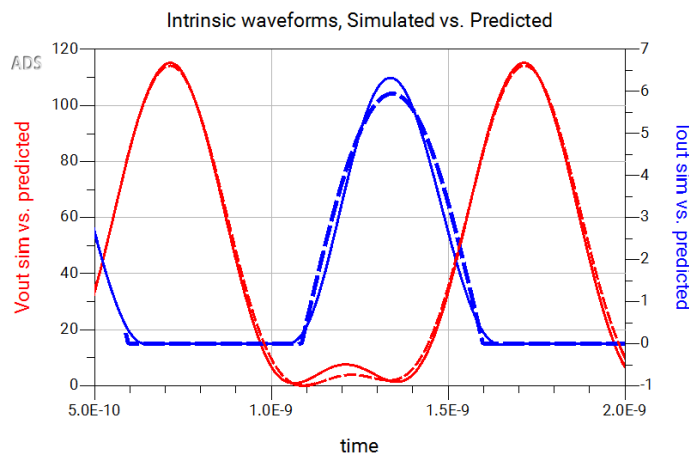


Figure 6. IV swing trajectories for alphas 0.0 (blue) and 1.0 (teal) and q point (black), calculated from DCIV sweep data.

Once the Nexus Connect task is complete, we achieve a good match between the simulated and predicted intrinsic volt-age and current waveforms, as well as the power and efficiency, as shown in Figure 7.

Extrinsic gammas, which are ultimately needed for matching network design, differ from intrinsic gammas because of the parasitics in the de-vice model. The Nexus Connect task sequentially optimizes the extrinsic gammas up to the fifth harmonic, see Figure 8, to minimize errors in achieving the target intrinsic gammas.

The entire sequence shown, comprising simulations, algorithms and decisions, is captured and executed in a single Nexus Connect flowchart. While an engineer could have initiated the same steps manually via the Nexus graphical user interface, it would be hard to reproduce the exact repeat-able steps from separate, disconnected analyses and tasks.



Pout_simulated	DrainEff_simulated
47.4	74.7
Pout_predicted	DrainEff_predicted
47.5	74.6

Figure 7. Simulated (solid) versus predicted (dashed) intrinsic waveforms, output power and efficiency.

Extending the Workflow with AI

Executable engineering workflows create structured interfaces that future AI tools can leverage. In AI-enabled design environments, agents can operate autonomously, while skills define repeatable sets of instructions that agents or large language models can invoke. A flowchart-based RF workflow makes this integration more practical because each step, decision and parameter is already formalized.

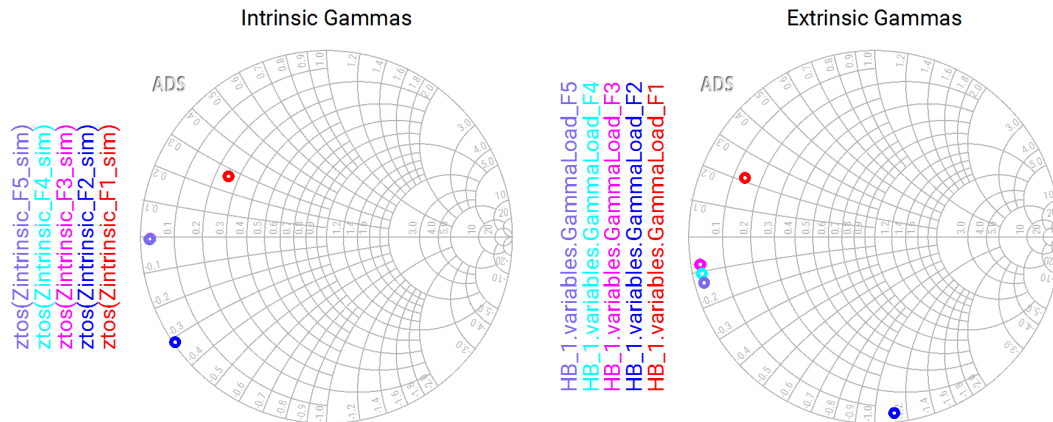


Figure 8. Intrinsic and extrinsic gammas at each harmonic.

For example, an AI agent could extend the PA workflow to characterize available transistors in a technology and identify the devices that best meet a target set of requirements. The agent could invoke a skill that describes how to run a DCIV sweep, calculate a Class J load line and evaluate candidate operating points. As AI tools mature, this type of structured workflow will help engineers delegate routine exploration while retaining control over design strategy.

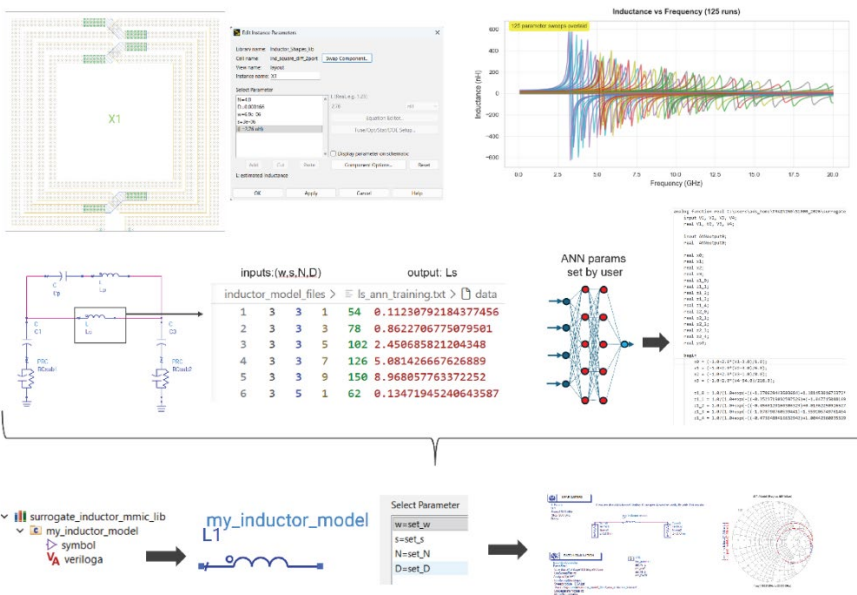


Figure 9. An ANN model of an inductor is validated and synthesized in a circuit via a Python script.

Surrogate modeling involves an upfront, sometimes resource-intensive training process that requires extensive characterization (for example, EM simulations). Once training is complete and validated, simulations and optimization using physical structures become ultra-fast, making surrogate models strong candidates for AI-enabled or fully autonomous design. Surrogate approaches have received considerable attention in recent years because they can synthesize non-traditional, QR-code-like physical structures with well-behaved frequency responses.⁴ Recent re-search has even demonstrated the potential of such structures for harmonically matching a PA, as this example requires.⁵ Note that surrogate modeling is not limited to pixelated structures; it can work for any parameterizable physical structure, even an inductor.

To illustrate, an ANN-based surrogate model was extracted for a set of inductors with varying geometries in a GaN MMIC process, as shown in Figure 9. First, EM simulations are performed using Keysight RFPro for each unique set of inductor geometry parameters, such as width, spacing, number of turns and diameter. Next, a modified pi network with 11 components is fitted to the EM data. Then, for each circuit element in the pi network, a separate ANN is extracted using Keysight's proprietary ANN engine, available as a Python function in ADS.⁶ The ANN produces Verilog-A functions that relate the physical parameters (w , s , N , D) to the circuit element values (capacitors, inductors and resistors). The result is a fast model that mirrors physical results and has input parameters that can be tuned or optimized.

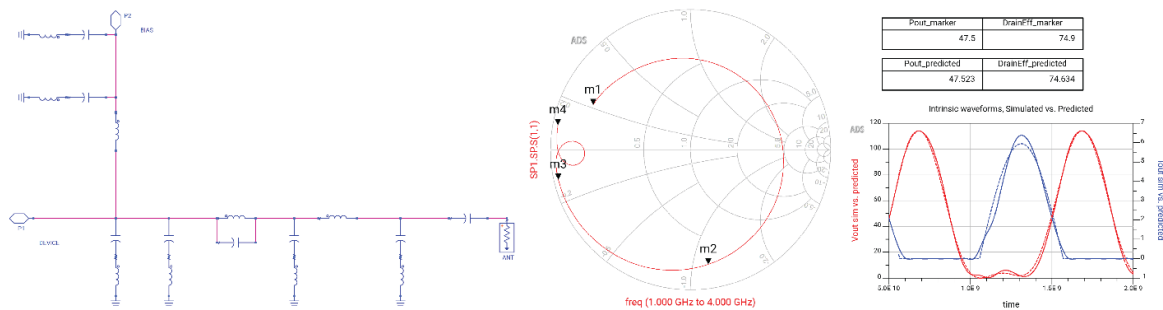


Figure 10. Output matching network tuned to Class J extrinsic targets using inductor surrogate models.

The surrogate model was then used to quickly tune and optimize an output-matching network that presents the proper external impedances to the device, yielding near-ideal Class J operation, as shown in Figure 10.

As more details of lower-level engineering processes are diagrammed and captured in Nexus Connect, routine tasks become increasingly automated, allowing engineers using AI tools to focus on higher-level design. The path also leads to natural language interfaces that can augment or replace GUIs and scripts.

Business and Engineering Impact

Structured, automated workflows deliver value beyond faster simulation. By capturing engineering methods, assumptions and decision criteria in a reusable format, they help organizations improve consistency, scale complex analyses, and transfer specialist knowledge across teams. This creates a more efficient engineering environment while establishing the structured foundation required for AI-assisted design and optimization.

Improved repeatability: Teams can rerun the same validated workflow using consistent models, settings, and decision logic. This reduces variability between engineers and projects, limits the risk of omitted steps, and makes simulation results easier to reproduce and verify.

Greater scalability: Parameter sweeps, optimization routines, and multi-step simulation sequences can be coordinated automatically. Engineers can explore larger design spaces and evaluate more alternatives without manually configuring and monitoring every simulation run.

Better traceability: Design assumptions, simulation choices, parameter values and decision criteria are captured within the workflow itself. This creates a transparent record of how results were generated, supporting design reviews, troubleshooting, verification, and future reuse.

- **Faster collaboration:** Visual flows make complex engineering processes easier to understand, review and share. Specialists can encode proven methods that colleagues can reuse, helping teams collaborate across disciplines and locations while reducing dependence on informal knowledge transfer.
- **Greater AI readiness:** Structured workflows expose clear interfaces between inputs, simulation tasks, results, and engineering decisions. This gives AI agents and optimization algorithms a controlled environment in which to automate tasks, interpret outputs and recommend next steps while remaining grounded in validated engineering processes.

Conclusion

Moving from ad hoc simulation to executable, flowchart-based orchestration can make RF design workflows more repeatable, scalable and transparent. The Class J PA example demonstrates how a complex sequence of simulations, algorithms and decisions can be captured in a single reusable process. By combining visual workflows with Python automation, engineers can reduce workflow friction while preserving flexibility and technical control. As AI-enabled modeling and agentic design tools evolve, these structured workflows will provide a foundation for faster optimization, improved collaboration, and more intelligent RF design automation.



As a mission-critical design enablement partner, Keysight connects design, emulation, and test so customers can solve the world's most complex engineering challenges. Accelerate innovation at www.keysight.com.

This information is subject to change without notice. © Keysight Technologies, 2026, Published in USA, September 9, 2026, 3126-1538.EN