Microwave Journal

Reimagining Possibilities for Next-Gen Simulation in RF EDA

Cedric Pujol and Matt Ozalas
Keysight Technologies, Santa Rosa, Calif.

Don Dingee
STRATISET, San Antonio, Texas

Historically, engineers procured CAD software with the implicit understanding that their hardware design process must conform to how a given software tool works. Vendors define their user interface and functionality, hoping to cover as much of the design process as possible. Engineering teams have a variety of tasks in their workflow and find or create tools for those tasks. Switching any tool in the suite usually means changing the steps required to complete a task, and sometimes the entire design workflow.

As hardware evolved in complexity, designers started to need multiple software tools to complete all the tasks in their workflow. Customers often settle on combinations of vendor and homegrown tools fitting their specific EDA needs, increasing the difficulty of adding or switching tools. Industry interoperability initiatives such as Si2 with OpenAccess implementations for EDA data repositories and APIs made tighter vendor-supplied integration and better growth paths feasible. Still, integrating RF simulators into modern workflows remains challenging, prompting a solution.

Keysight researchers are imagining a different direction for an all-new RF simulator suited for complex high frequency RF design, retaining integration benefits with a new programmatically-controlled, extensible platform. This research deepens the ties between measurement and simulation technology. Keysight explores the motives for developing a next-generation RF simulator, how designer productivity is shifting with help from research insights and the possibilities this simulator offers in automated RF design workflows, including roles for AI.

MOTIVES FOR PROGRAMMATICALLY CONTROLLED RF SIMULATION

Software development teams seek to combine and prioritize customer needs and align them with research and development knowledge, a product's inner implementation secrets and the resources needed to effect changes. If a software tool finds widespread adoption, it becomes difficult to simultaneously morph into different design spaces while maintaining a state-of-the-art core engine. Technological expansion may suggest advanced capabilities, but required engine modifications limit or preclude adding them without substantial risk or costs. At some point, breaking with the past and creating an all-new engine architecture and code base becomes a better option for developers and users, similar to how buying a new computer can be more cost-effective than upgrading.

An RF simulator, the engine driving any modern RF EDA workflow, is no exception. Keysight's RF simulation technology evolves in lockstep with its measurement science. That philosophy helped Advanced Design System (ADS) become a key platform for complex RF design with extensive circuit electromagnetic (EM) co-simulation capability.

Still, more is possible. Reimagining the possibilities for next-generation simulation in RF EDA began with the realization that Keysight's previous RF simulator engine is sophisticated but traps an immense amount

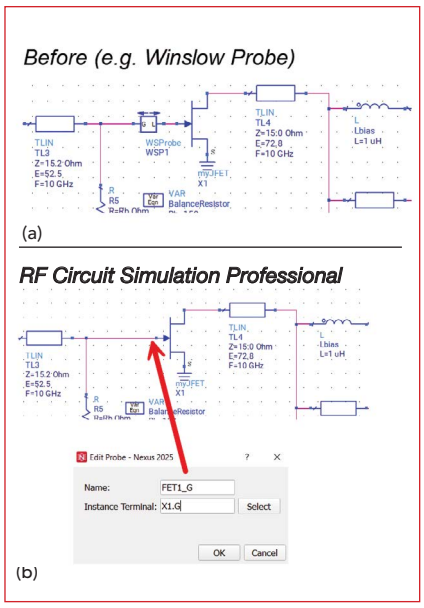


Fig. 1 Schematics of (a) a Winslow probe inserted directly versus (b) virtual insertion via the user interface.

of information behind its interface, rendering it inaccessible. Keysight researchers, inspired by measurement workflows, asked what might be possible by re-architecting a streamlined RF simulator, creating more hooks for programmatic execution control, leveraging interstitial data and matching features to real-world engineering design processes. In this light, a simulation engine becomes more like a measurement instrument: portable, automatable and pluggable into many different workflows. This curiosity reveals several motives for a high-stakes project.

- **Modernizing the user interface and uncluttering schematics:** User interface design directly affects user perceptions of ease of use. Powerful tools would start simulations directly from the schematic interface and avoid the creation of spin-off schematics to handle specific simulations. Layout versus schematic (LVS) verification could be more straightforward; the previous RF simulator inserts simulator controllers, probes (see ***Figure 1***) and other info on a schematic, requiring stripping that information back out for error-free verification.
- **Providing methods to control user interfaces and simulation invocation:** Python scripts calling an API are a popular addition to the previous simulator. Still, more opportunities exist to expand and customize user interfaces and introduce standard Python analysis functions. There is also the possibility for command-stream invocation of simulations, even bypassing the simulator user interface (headless), with calls issued by ADS or other workflow tools.
- **Separating simulation analysis from workflow tasks and speeding up analyses:** Semantics aside, simulation has two distinct steps: picking an analysis, like S-parameters, envelope or harmonic balance, and setting up a task, which is a sequence of analyses, like sweeps or optimization. In short, an analysis defines behavior evaluation, and tasks control how analyses execute (see ***Figure 2***). Once that distinction is made, more efficient parallelization and significant speedups become possible.
- **Pre-processing for defining sweeps, optimization and visualization:** Fine-grained parameter control of simulations via a user interface is necessary, but is not enough to make simulations more efficient. Users commonly apply post-processing for data display to extract and visualize subsets of data (for instance, zooming in on a range of frequencies), yet the simulation runs across the entire data set. Pre-processing could define reusable expressions evaluated during simulation, in analyses or tasks, with enhanced Python filtering capability for results (see ***Figure 3***). Automatically tying simulation pre-processing to optimization and plotting could also save more steps in the workflow.
- **Capturing and sharing simulation, display and optimization parameters:** Control of numerous simulation parameters is a plus, but it also introduces a potential traceability issue. Reproducing simulations run yesterday, last week or last month with the exact same parameter settings can be daunting. In the measurement domain, routines embedded in instruments provide reproducible specification compliance testing. A similar approach introduces Performances, which capture settings, expressions, filters, optimizations and specifications for traceable, reusable and sharable simulations.
- **Unlocking 10x better optimization speed with parallelization:** Although good algorithms exist in the previous RF simulator, optimization remains challenging as frequencies rise, heterogeneous technologies integrate and specifications tighten. A significant hurdle is shifting from individually optimized subsystems to larger problems in the global optimization of an entire design. An example is band filters, with perhaps ten or more working together, and leakage from one band can affect the performance of others. Another could be optimizing on different corners to find values to minimize the impact of process impairments. Joint optimization increases the domain space and the multi-dimensional surface for optimization. It is also possible

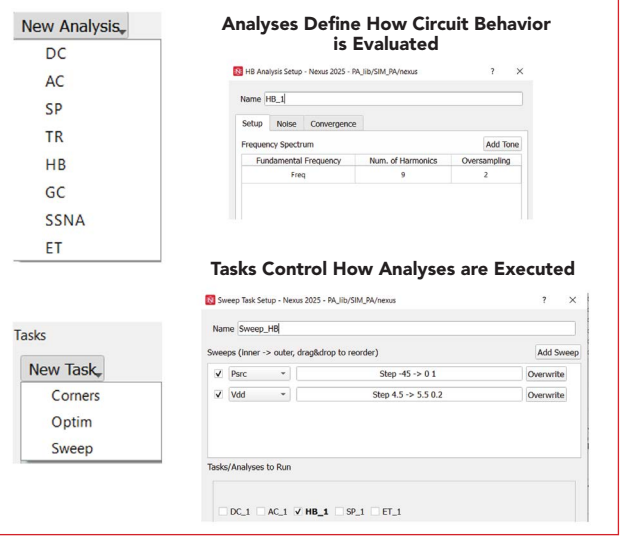


Fig. 2 Separation between analyses and tasks in the simulator user interface.

that, in a heterogeneous scenario, individual subsystems come from different design platforms, making global optimization more difficult. New algorithms and multi-threaded parallel execution could boost execution speed by an order of magnitude.

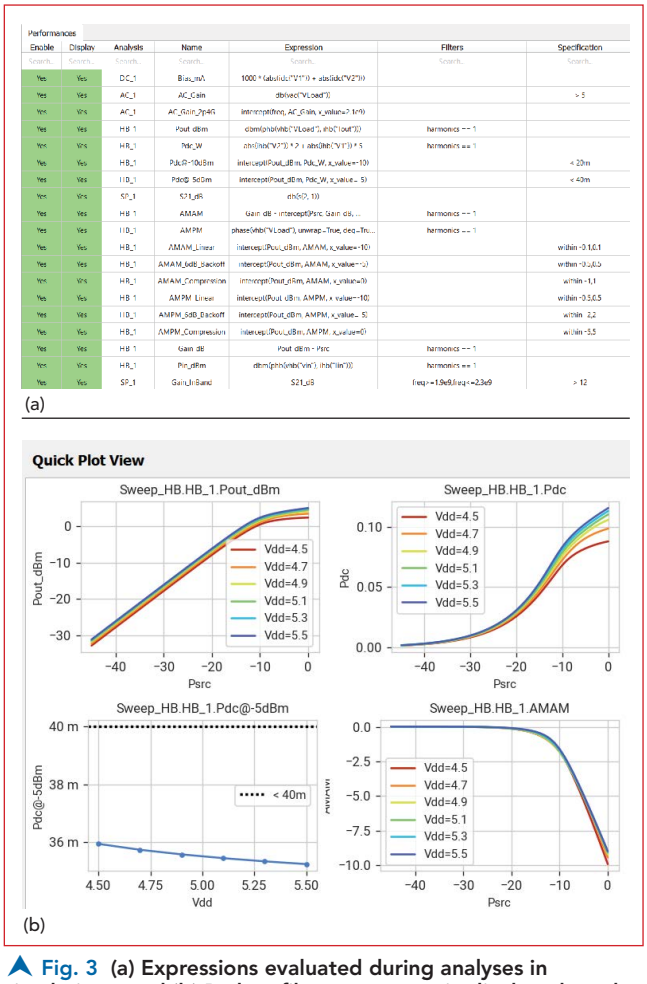
- **Setting up flexible licensing for immediate team access to features as introduced:** Licensing has also held back simulation users in some situations. An industry trend recognizes different lifecycles. Teams desire more stable design platforms with updates once, twice or maybe three times a year. RF designers tend to want on-demand simulator updates within weeks or even days of fix and feature releases. A modular simulator that can update without changing the underlying platform solves some issues. Flexible licensing that allows using any simulator feature without relicensing, including the latest feature releases, would be an improvement as design workflows morph.

Of course, these motives coexist with ongoing efforts to add and enhance simulator analysis types and improve their raw execution speed. For instance, work continues on gain compression enhancements and a revamped memory model for effects on high bandwidth signals. The primary mission of the RF simulator remains to deliver accuracy but not sacrifice time. Addressing these motives for programmatic control unlocks another level in workflow productivity while enabling intense analysis of complex RF designs.

ENHANCING DESIGNER PRODUCTIVITY AROUND RF SIMULATORS

Some of these productivity gains are achievable in Keysight's previous RF simulator, with notable differences between Keysight and non-Keysight platforms. The next-generation RF simulator, RF Circuit Simulation Professional, further expands the possibilities for users of various design and layout platforms with its modularity. Design teams now get the same core RF simulator capability regardless of their platform choice.

Allowing design teams more access to the RF simulator opens new solutions to complex problems, even solutions that teams may have avoided or worked around before in a "good-enough" workflow with some compromises, assumptions or best guesses. Four examples illustrate what is immediately possible with programmatic control in an API-enabled design platform.



▲ **Fig. 3 (a) Expressions evaluated during analyses in simulations, and (b) Python filters on curves in displayed results.**

Adapting access to simulations with Python

Many users are familiar with Python scripting for basic user interface customization, data visualization and oft-repeated format conversions. Rapid development in the Python ecosystem introduces powerful capabilities that designers can easily stitch into their workflow. Adding post-processing functions and using them as if they were built-in simulator functions is a new capability. For instance, designers can remove dozens of equations in the simulator's Data Display and replace them with a single, reusable, upgradable SciPy function.

More possibilities open when the RF simulator stops being a black box and offers Python control over execution. Many users stick with Keysight's optimizers, but others develop their own for specific problems. Bolting in an external optimizer has meant months of joint customization work between Keysight application and R&D teams, designers and any third parties involved in developing the algorithm. Now, using a third-party optimizer to access the RF simulator's API via Python scripts and a wrapper becomes seamless. AI-written Python code offers another level of productivity gains.

Tying simulations to areas of circuit layouts for troubleshooting

Designers celebrate simulation success. However, experience says things don't go smoothly the first time through a design. Bumps in the flow become more common when subsystems come from various sources, some designed by third parties, such as in many 3DHI projects. Designers have their layout and less-than-good simulations, but where in the circuit or layout are the problems and what fixes can make them go away?

Troubleshooting is perhaps more crucial in a workflow than efficient simulation because it consumes time and drives schedules out of control if not solved quickly. Answering the question becomes straightforward when information previously hidden in the simulator is visible. In ADS, designers can see highlighted layout segments contributing to any out-of-

bounds simulation results. Instead of guessing where the problem might be, they can dive directly into a specific spot to fix it.

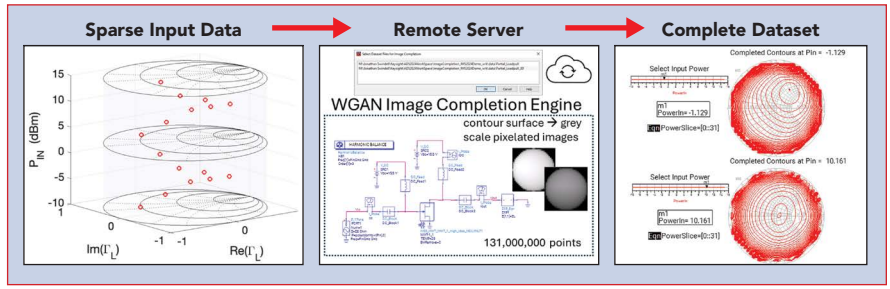
Gaining workflow flexibility through greater access

Seeing inside structures has granular benefits. Modifying a physical layout typically requires a labor-intensive manual sequence of steps, making it challenging to create perturbations essential for many model training workflows. With access, a set of API commands can alternatively represent any physical layout or schematic, enabling modification of all the information in the design programmatically, as if the physical design were a Python function. To illustrate this, Keysight developers devised a utility that extracts an API-based Python script from a physical design or schematic — running the script recreates the schematic or layout accurately. From there, a human or AI coder can manipulate the design by changing the function, rather than the physical design itself.

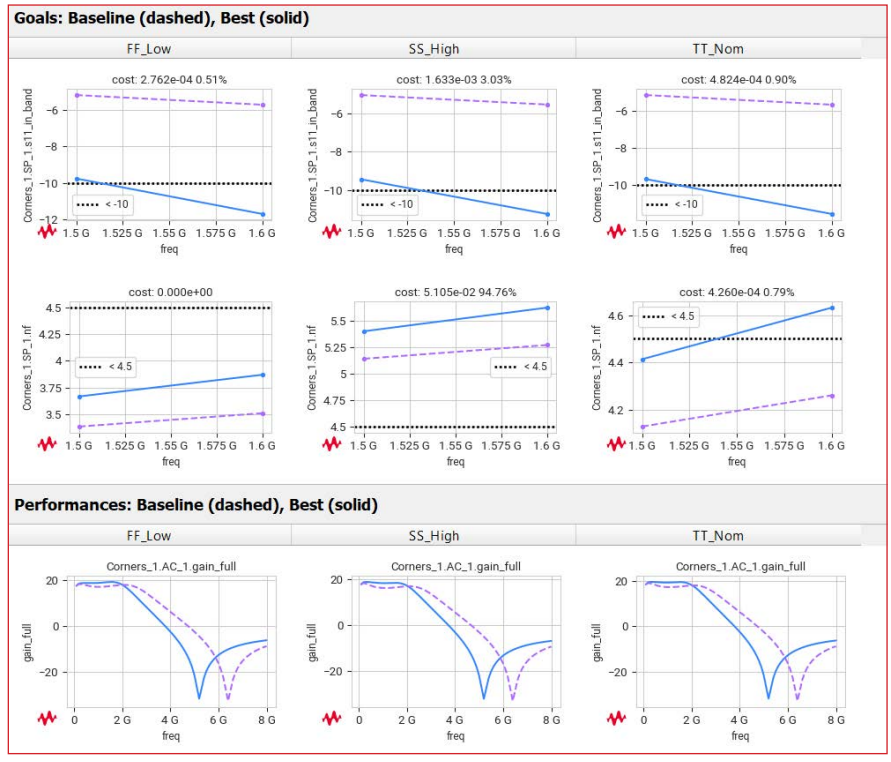
Functional automation provides a pathway to decrease simulation or measurement time by pretraining surrogate models. Researchers at Baylor University recently illustrated the concept using load-pull data.[1] Their approach uses an automated simulation workflow built on a functionalized schematic to generate contours for various conditions on an MWT-1 transistor. Upfront simulation collects 131 million points to create images that train an AI engine using a Wasserstein Generative Adversarial Network (WGAN). The pretrained model derives complete simulation or measurement contours from an extremely sparse data set (see ***Figure 4***), using 1500x fewer points. Pretraining surrogate modeling approaches can potentially capture EM and multiphysics effects, cutting physical design time.

Enabling parallel optimization and visualization

Bringing all optimization capabilities into RF Circuit Simulation Professional with any imported Python extensions means all the algorithms, parallelization and visualization tools are available on any



▲ **Fig. 4** Load pull contours derived from a pretrained model. *Source:* Jonathan Swindell, Baylor University.



▲ **Fig. 5** Live optimization compares simulation progress to goals, showing algorithm effectiveness.

design platform. Teams on different platforms can also share their optimization setups seamlessly.

Live visualization progress (see ***Figure 5***) also allows teams to see if and how results converge during simulation runs, determining at a glance if the algorithm is effective or needs changing. Optima feed back into the design on the host platform automatically. These visualizations are helpful in several scenarios. One is employing live retrieval of measurements. Another is where system metrics such as error vector magnitude (EVM) are in play, where it's crucial to avoid overdesign triggered by approximations. A third case is analyzing process variations to enhance manufacturing yield.

POSSIBILITIES FOR AUTOMATED RF DESIGN WORKFLOWS

The biggest wins for design teams deploying programmatically controlled simulation in RF EDA workflows may be in the near future. The first logical outcome is an optimizer ecosystem, with more RF researchers participating. Innovative third parties or in-house research teams can identify and mechanize new algorithms in familiar Python code (see ***Figure 6***). Additionally, a new business model will likely emerge for optimization add-ons that are marketable to customers on many platforms, except for previously required and lengthy platform customization cycles.

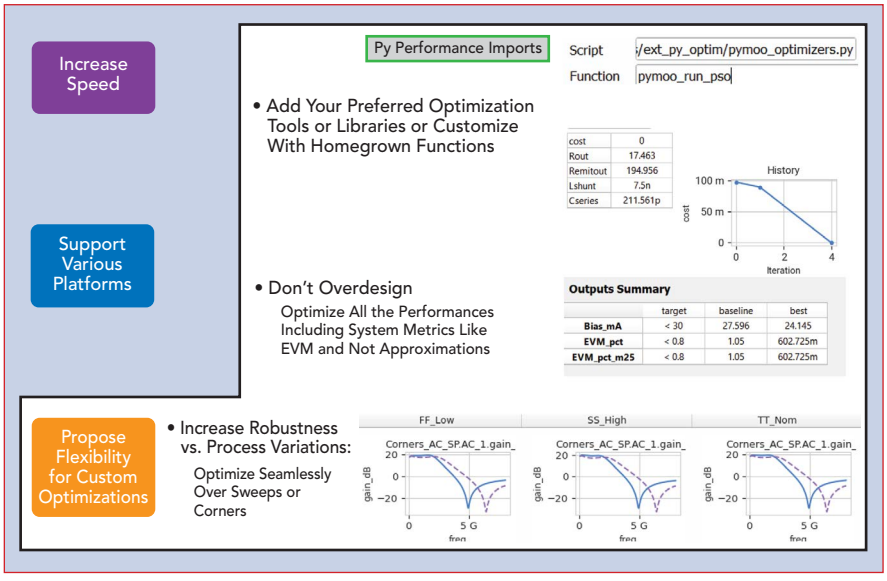


Fig. 6 **Revamped optimization sets the stage for an optimizer ecosystem.**

The following two possibilities exist in a post-Python space where customization becomes more intuitive. Keysight is working on a no-code solution for sequencing simulations using a visual, flowchart-like solution that captures requirements and workflow in a single diagram. A defining feature of this approach is conditional execution, which can adapt workflows on the fly based on simulation results. Designers will not need to understand the API syntax to make a sequence work, although using Python calls to the API remains an option. This no-code capability would make it easier for less experienced designers to use the simulator and for design teams to share their simulation knowledge.

Finally, there is AI. Exposing simulator control sets the stage for AI training, as mentioned previously. It also allows natural-language queries from an AI front-end to run simulations in a design workflow or call a specific design exploration ad-hoc. Imagine sitting in a design review or a customer meeting, fielding a question not answered in the presentation, calling the simulator and refining the prompts until the answer appears within minutes. Combining accuracy, speed and control puts RF simulation within reach for more designers, and the outcomes will be stunning as better first-pass design success and more realistic virtual twins help manage burgeoning RF system complexity. ■

References

1. Swindell et al.,"Multi-dimensional Load-Pull Extrapolation for Accelerated Computer-Aided Design (CAD) Simulations," *USNC-URSI NRSM*, January 2025.