

WHITE PAPER

# Bridging Wireless Communications Design and Testing with MATLAB

## Contents

Key Requirements for Today's Wireless Research and Development.....	3
Working with Standard-Compliant Systems and Signals.....	4
Integrated Environment for RF-Digital Modeling and Simulation.....	5
Over-the-Air Testing with Software-Defined Radios and RF Instruments.....	5
Accelerating Wireless System Prototyping.....	6
Wireless System Design with MATLAB.....	7
Simulation of Wireless Communications Standards.....	8
Case Study: 5G NR Downlink Throughput Simulation.....	10
Learn More About Standard-Compliant Simulation and Design.....	11
Joint RF-Baseband Modeling and Simulation.....	13
Case Study: Modeling Massive MIMO Antenna Arrays and Beamforming.....	14
Learn More About MIMO Processing and Beamforming in Wireless Systems:.....	16
Case Study: Power Amplifier Characterization with DPD for Reduced Signal Distortion.....	16
Learn More About Joint RF-Digital Modeling and Simulation.....	18
Design Verification and Testing.....	18
Standard-Compliant Signal Generation and Measurement.....	18
Over-the-Air Testing with RF Instruments and Software-Defined Radios.....	19
Case Study: LTE Scanner with SDR Hardware.....	20
Learn More About Testing with RF Instruments and Software-Defined Radios.....	21
Wireless Prototyping and Implementation.....	22
Prototyping Workflow for FPGA and SoC-Based SDR Platforms.....	22
Case Study: Optimized HDL Code for FPGA and ASIC Implementation.....	23
Learn More About Wireless Prototyping and Implementation.....	25
Summary.....	25
Learn More About Wireless Design with MATLAB.....	25
See How Leading Companies Are Using MATLAB for Wireless Design.....	25

Advances in wireless communications require a new level of technology integration, stretching the old ways of doing development to the limit.

This white paper examines key challenges in the changing landscape of wireless system design that affect the speed and success of product development.

With current technology, the entire signal chain from RF to baseband can be implemented in a single programmable device or module. Consider the expertise required to use, let alone design, one of these devices: RF, digital logic, DSP, embedded software, and system architecture. And to integrate it into a complete system, you'll need to know even more: antenna design, propagation, and one or more wireless standards.

For a typical R&D group, it can be difficult to prototype or validate new design concepts, particularly multi-antenna systems and highly coupled digital and RF front ends. This is major challenge for R&D teams as they race to deliver the next breakthrough product. The days of domain experts working separately, using separate tools, are numbered.

Fortunately, there is a better way. Wireless engineering teams can benefit from a software environment that connects algorithm design, system simulation, over-the-air testing, prototyping, and implementation. Compared with groups still designing in silos, teams taking advantage of earlier design integration report saving as much as 30% in overall development time and 85% in functional verification time.

## Key Requirements for Today's Wireless Research and Development

Transforming a fragmented wireless development workflow calls for tools that can integrate multiple engineering disciplines into a coherent workflow (Figure 1). The ideal workflow should accelerate delivery of working prototypes and products by enabling engineering teams to:

- Create standard-compliant waveforms and reference models to verify designs at each stage
- Design baseband, RF, and antenna components in the context of an end-to-end simulation
- Prototype and validate designs using RF instruments and off-the-shelf software-defined radio (SDR) hardware

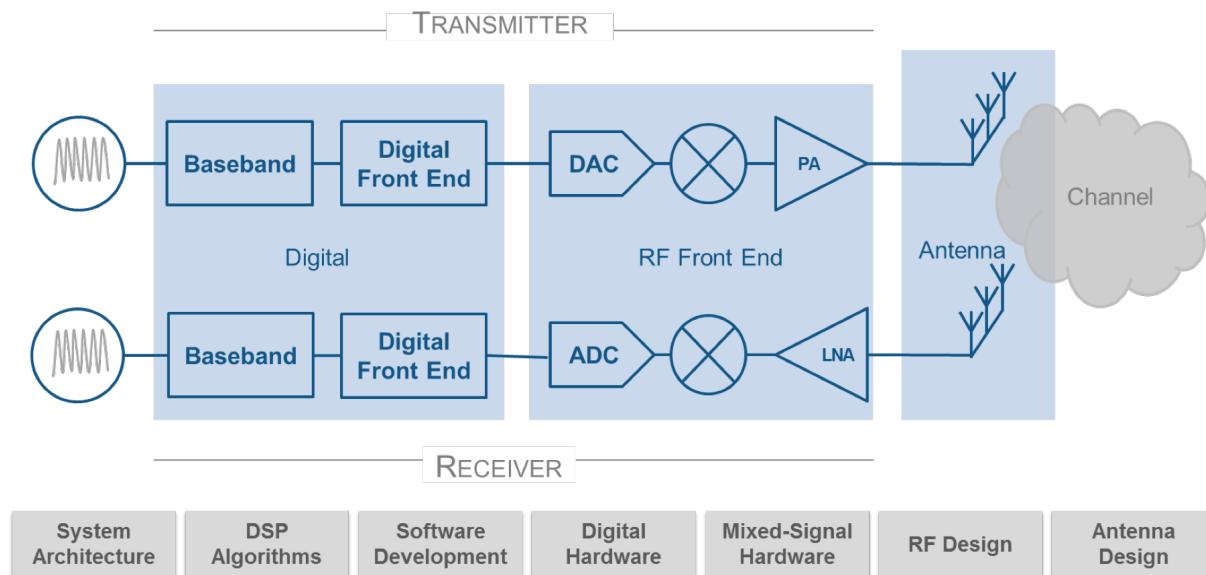


Figure 1. The many engineering skills required for development of today's wireless systems.

## Working with Standard-Compliant Systems and Signals

Efficiently confirming standard compliance of 5G, LTE, and WLAN products requires tools that can meet the different requirements at each stage of research, design, and testing.

Researchers often develop in-house simulation tools to explore algorithms and run performance simulations.

Typically, these tools are poorly documented and difficult to use for product development tasks such as prototyping and testing. Design engineers and test engineers may use standard-based waveform generation and analysis software supplied with test and measurement equipment. This software cannot be used independently of the test hardware and is difficult to modify for scenarios that fall outside the original menu of parameters and options. In addition, with dedicated test software it is difficult to access the intermediate signal information needed to debug failures.

When different tools are used for simulation and hardware testing, additional work is needed to port the simulation tests into the hardware testing environment. This work is time-consuming and can introduce inconsistencies that are challenging to debug and correct.

This process can be improved by using the same software at all stages of the development process, spanning four common use cases (Table 1).

Use Case	Engineering Goals
End-to-end simulation	<ul style="list-style-type: none"> <li>Assess how design choices affect system performance</li> <li>Confirm that the system complies with the standard</li> </ul>
Standard-compliant waveform generation	<ul style="list-style-type: none"> <li>Test designs with synthetic or live over-the-air signals</li> </ul>
Golden reference for design verification	<ul style="list-style-type: none"> <li>Verify that the design works as it should</li> </ul>
Signal reception and recovery	<ul style="list-style-type: none"> <li>Decode real-world signals</li> <li>Validate receiver designs</li> </ul>

Table 1. Use cases for assuring software compliance with wireless standards.

To span this set of tasks, the software should have these attributes:

- Product architecture and programming interface that operates at three levels:
  - Low-level functions (e.g., modulation, mapping, precoding) with open interfaces
  - Mid-level functions that process a complete link (physical channels and signals) in one step
  - High-level signal generation functions and apps
- Open implementations using a high-level programming language to facilitate understanding, experimentation, and customization
- Hardware-agnostic testing interfaces, so the generated signals and the test bench can be used for simulation and test hardware independent of a specific manufacturer

## Integrated Environment for RF-Digital Modeling and Simulation

Higher data rates, massive connectivity, lower power consumption, and other ambitious goals for wireless systems can only be achieved by combining advanced digital, RF, and antenna technologies. Traditionally, each of these components has been designed separately, only to be integrated, tested, and debugged after the first hardware prototype is built.

If you design baseband algorithms without considering RF impairments, they're unlikely to work in the real world. If you're an RF front-end designer, the DSP and digital control algorithms and antenna configuration will affect system performance and cost. You need to know how RF and DSP are going to work together, because that will affect overall system performance and cost. When you are using multiple tools from different vendors, it is extremely difficult, expensive, and slow to test, debug, and correct errors, leaving little or no time to optimize the design.

A flexible, integrated simulation environment provides critical advantages for wireless system design. It enables domain experts in each area to use the most appropriate tool and contribute to the design and verification of the complete wireless system.

The models can be configured to achieve goals such as simulation speed or fidelity to the target implementation, and custom functions can be easily added. This approach is particularly useful for complex applications such as:

- Modeling multi-antenna (MIMO) systems used in 5G, LTE-Advanced, and WLAN systems, including antenna arrays, propagation patterns, and beamforming
- Simulating end-to-end communication links, including RF and channel impairments and algorithms to mitigate those impairments
- Modeling digitally controlled RF front ends and transceivers, providing fast, accurate simulation results and control over parameters that are difficult to tune and debug in the hardware
- Exploring, designing, and analyzing techniques to improve throughput, efficiency, and robustness, including deep learning and machine learning

## Over-the-Air Testing with Software-Defined Radios and RF Instruments

Ultimately a wireless system must transmit and receive signals over the air. With the commercial availability of software-defined radio hardware, it is now practical and cost-effective to perform over-the-air tests and capture live signals much earlier in the development process. To date, the challenge has been in the software.

The first generation of SDR development tools fell into two categories: open source frameworks and proprietary software for specific vendors' hardware. In both cases, the SDR development is divorced from the algorithm development and simulation phase. It is now practical to connect simulation and algorithm development software to SDR hardware. This approach enables reuse of models and waveforms to verify designs with live signals, saving significant time and cost.

For test and measurement applications requiring calibrated RF instruments, engineers have typically relied on software provided with the instruments. This software is designed to perform specific tasks on specific hardware. For these applications, it can also be significantly more efficient to use simulation and algorithm development software to perform signal generation and analysis tasks.

The ideal software is hardware agnostic, so engineers can:

- Reuse signal generators, measurements, and test benches to ensure continuity from the simulation stage to live testing to production test and verification
- Work with a range of off-the-shelf hardware to satisfy different requirements and budgets

## Accelerating Wireless System Prototyping

Wireless engineers increasingly rely on early hardware prototypes and proof-of-concept testbeds using FPGA or programmable SoC hardware. While prototyping platforms are available, wireless system engineers and research teams frequently lack the hardware design expertise to develop and debug those prototypes.

Many engineering teams are turning to Model-Based Design, which provides a cohesive workflow that uses models to:

- Create an executable golden reference specification
- Develop and verify fixed-point hardware-ready designs in simulation
- Automatically generate VHDL®, Verilog®, and C code
- Automate the workflow for prototyping and verification on FPGA or SoC hardware

With Model-Based Design (Figure 2), models enable exploration and validation of hardware architectures, automatic code generation to reduce implementation errors, and continuous verification to decrease the time spent in testing and debugging software and hardware. Figure 3 shows that using Model-Based Design can reduce implementation and verification time of HDL IP for wireless ASICs by 60% or more.

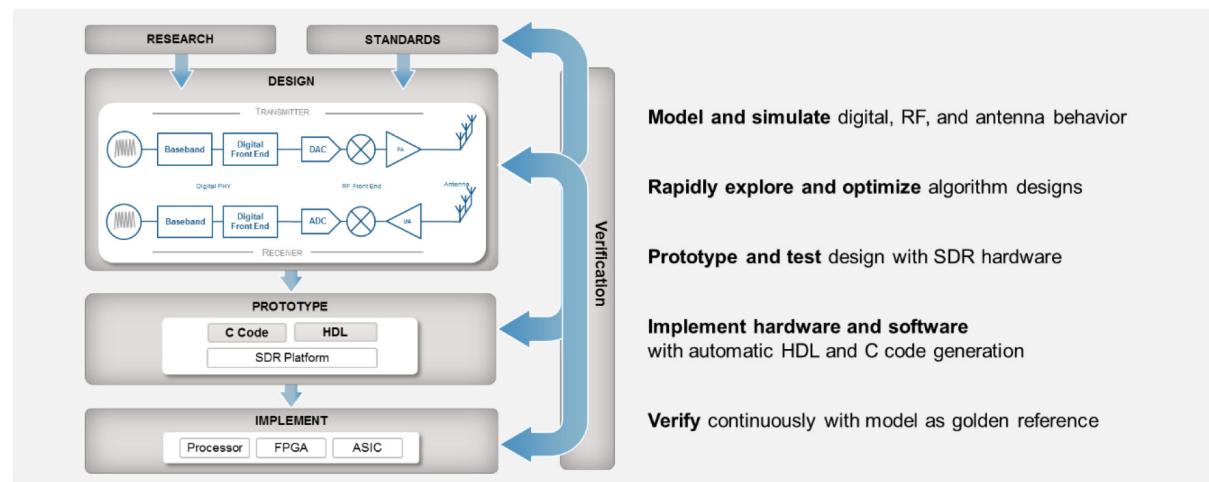


Figure 2. Unifying and accelerating the wireless development workflow by using Model-Based Design.

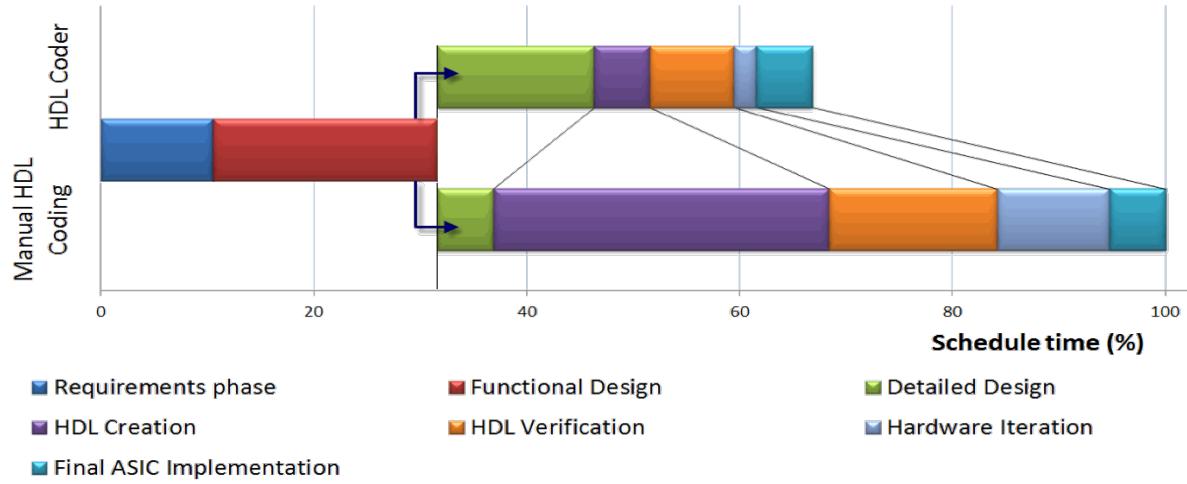


Figure 3. Time savings using Model-Based Design to develop HDL IP for wireless ASICs.

## Wireless System Design with MATLAB

Wireless engineers rely on MATLAB® to develop algorithms, analyze data, explore new technologies, and publish thousands of research papers and software contributions. The reason is that MATLAB is ideal for physical layer modeling, the foundation of all wireless systems.

Many leading companies now use MATLAB for much more than algorithm development and analysis. Their engineering teams take MATLAB algorithms all the way to full system simulation and implementation of LTE, WLAN, 5G, and other wireless systems. These engineers use MATLAB and Simulink® to integrate RF, mixed-signal, and digital technologies into multidomain system models, connect to hardware for over-the-air testing with live signals, and prototype and implement their designs on a range of hardware (Figure 4).

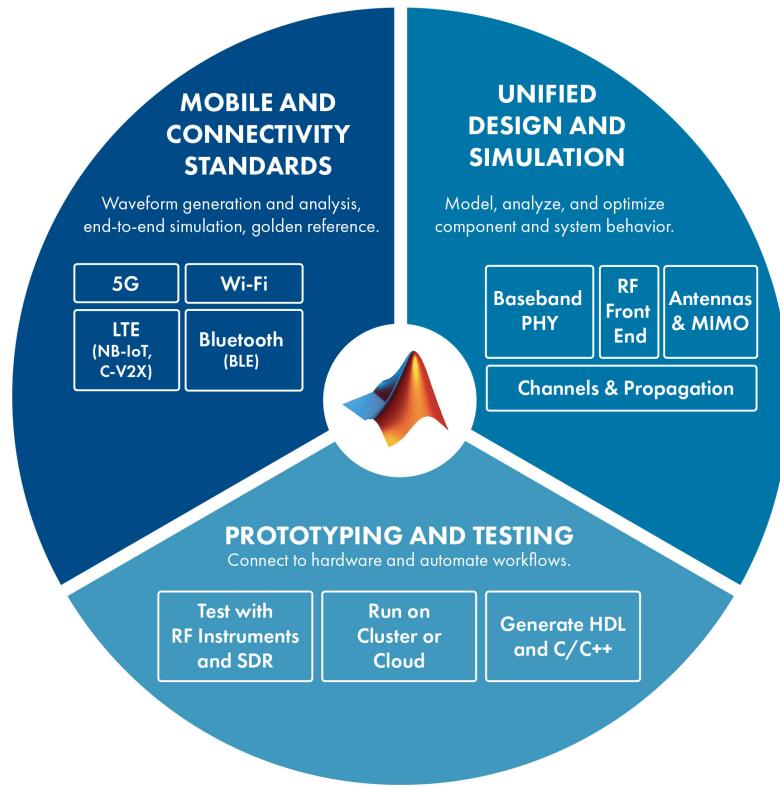


Figure 4. Streamlining development of wireless systems by using the unified environment of MATLAB and Simulink.

As a result, they eliminate steps and deliver working designs faster because they can:

- **Prove compliance with standards** in simulation and over-the-air tests
- **Explore and optimize system designs** with joint baseband-RF simulations
- **Eliminate design problems** before moving to implementation
- **Streamline testing and verification** with simulation and over-the-air tests
- **Automatically generate HDL or C code** from algorithm models
- **Reuse models** to speed up design iterations and next-generation projects

The following examples and case studies illustrate many of the ways that MATLAB and Model-Based Design address pervasive design challenges in wireless development.

### Simulation of Wireless Communications Standards

As the wireless industry moves toward ubiquitous wireless access, LTE, WLAN, Bluetooth®, and other standards will continue to evolve and play an important role in future systems (Figure 5).

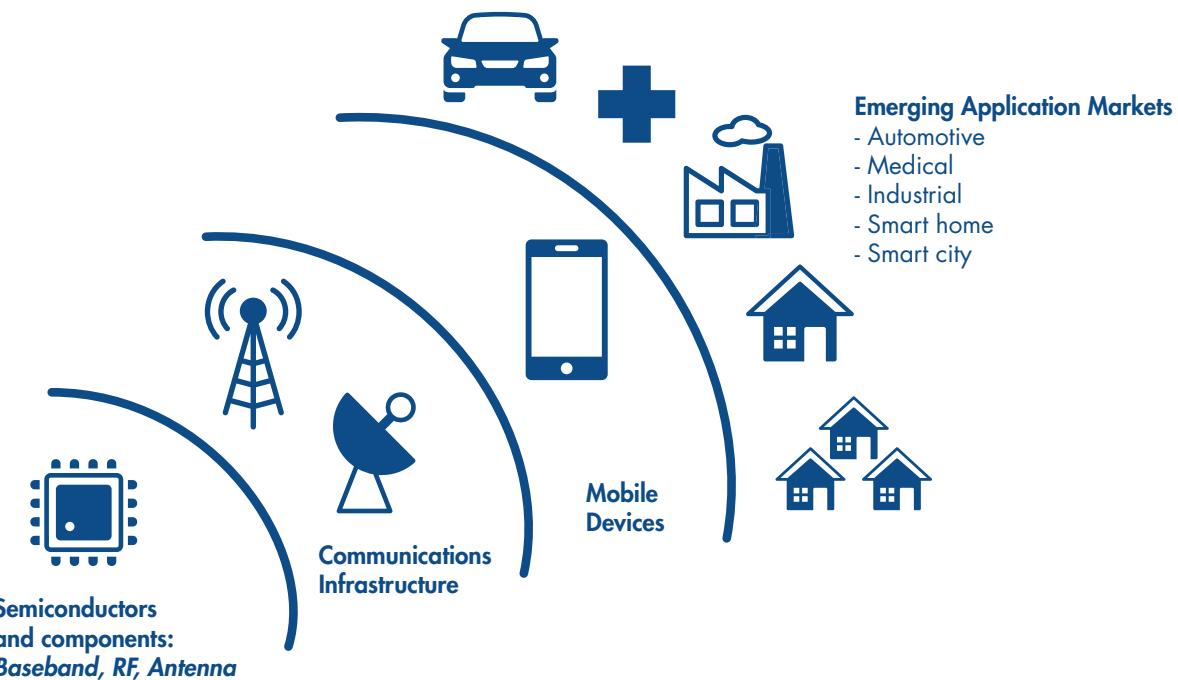


Figure 5. Growing connectivity in new applications enabled by the latest wireless standards and technologies.

A range of MATLAB toolboxes are available that enable engineers to design and test standard-compliant 5G, LTE, WLAN (802.11), and Bluetooth Low Energy systems. These toolboxes provide capabilities that support the four use cases outlined in the previous section.

Use Case	Capabilities of Wireless Standards Toolboxes
End-to-end simulation	<ul style="list-style-type: none"> <li>• Simulate the transmitter, channel models, and reference receiver</li> <li>• Analyze bit-error rate (BER) and throughput</li> </ul>
Waveform generation	<ul style="list-style-type: none"> <li>• Generate standard-compliant waveforms</li> <li>• Set parameters graphically or programmatically</li> <li>• Use in simulation and over-the-air with RF instruments and software-defined radios</li> </ul>
Golden reference for design verification	<ul style="list-style-type: none"> <li>• Customize “white box” implementations of standard-compliant algorithms</li> </ul>
Signal reception and recovery	<ul style="list-style-type: none"> <li>• Capture and decode real-world signals</li> <li>• Validate receiver designs</li> </ul>

These capabilities enable you to measure the impact of different algorithms, design choices, and channel conditions on system performance and throughput, develop efficient receiver algorithms, and verify that your system design conforms to the standard specification.

You can configure, simulate, measure, and analyze end-to-end communications links that conform to the 5G, LTE, 802.11 WLAN, or Bluetooth Low Energy standards. MATLAB toolboxes provide examples to help you explore baseband parameters and simulate the effects of RF designs and interference sources on system performance.

Because toolbox examples and functions contain open MATLAB code, you can use them to learn how the standards operate and as customizable reference models for developing and validating your proprietary implementations.

### Case Study: 5G NR Downlink Throughput Simulation

While there are many commonalities between 5G NR and LTE, starting with the names of most physical channels and signals, there are also significant differences between them.

5G NR is defined to support more use cases than LTE, and the latency requirement is much more stringent. This leads to several key changes in the physical layer for 5G NR. Among those changes:

- Carrier frequencies for 5G can be as high as 60 or 70 GHz, whereas LTE is only deployed below 6 GHz. Here again, this has significant consequences on the design of the physical layer, as beamforming becomes required to support those higher frequencies.
- At those higher frequencies, more spectrum is available, and 5G NR is set to take advantage of this spectrum with up to 400 MHz of bandwidth, and possibly more later.
- The subcarrier spacing, fixed in LTE at 15 kHz, can now take values between 15 and 240 kHz.

Figure 6 represents the end-to-end simulation of a 5G NR downlink shared channel (DL-SCH). This is the channel that carries user data. It also carries other pieces of information, such as the different types of system information blocks (SIBs).

The coding chain includes the usual steps such as CRC, code block segmentation, rate matching, and concatenation, all steps we are familiar with from LTE. The main difference with LTE is the use of LDPC coding. The output of the coding chain is a codeword. The coded data is then mapped to the physical downlink shared channel (PDSCH), modulated, and transmitted through a channel model. The receiver reconstructs the waveform after synchronization, demodulation, and channel estimation.

In Figure 6, the variable subcarrier spacing is visible in the plot on the left, and throughput measurements are shown on the right.

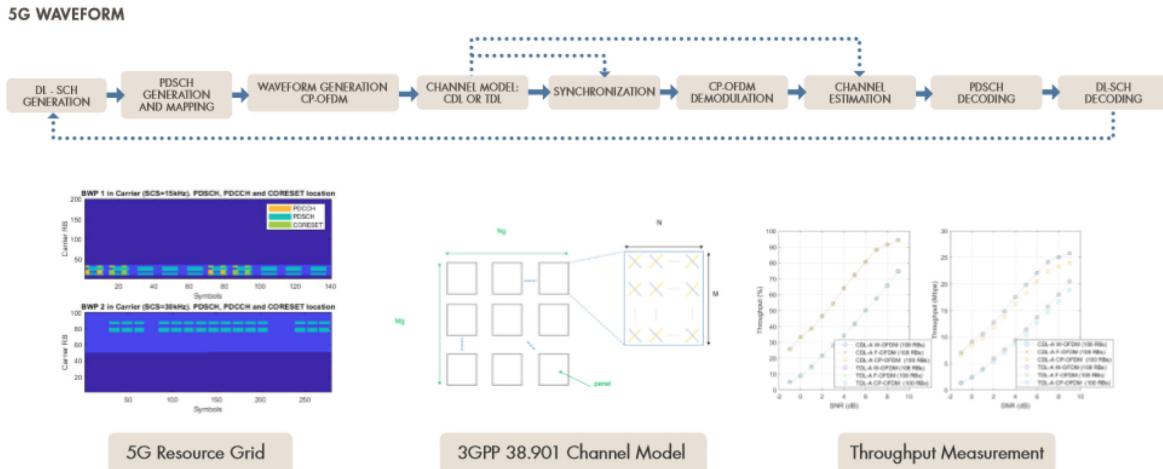
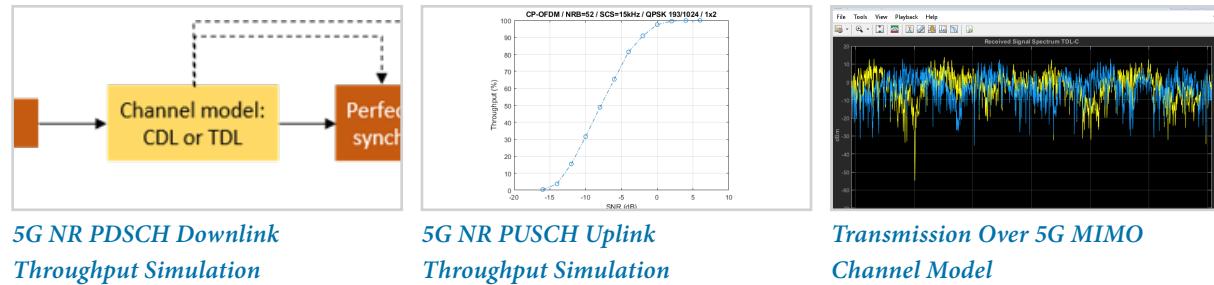


Figure 6. End-to-end simulation of a 5G NR downlink shared channel.

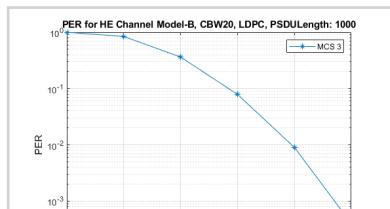
## Learn More About Standard-Compliant Simulation and Design

### 5G End-to-End Simulation

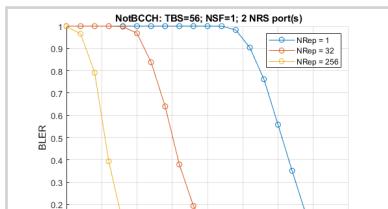


## LTE, WLAN, and Bluetooth End-to-End Simulation

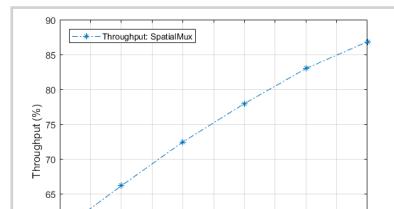
MATLAB toolboxes provide end-to-end simulation for LTE, WLAN, and Bluetooth wireless standards as well as 5G. These examples show you how to perform simulation at the link level (one transmitter and one receiver) and system level (multiple nodes).



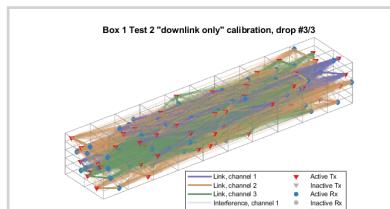
**802.11ax Packet Error Rate Simulation**



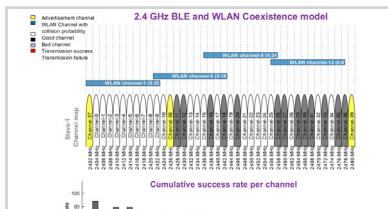
**NB-IoT Downlink Block Error Rate Simulation**



**LTE Throughput Conformance Test**



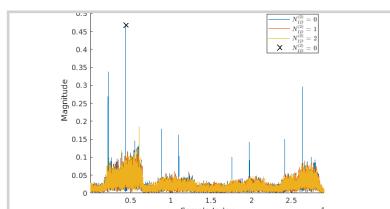
**802.11ax PHY-Focused System-Level Simulation**



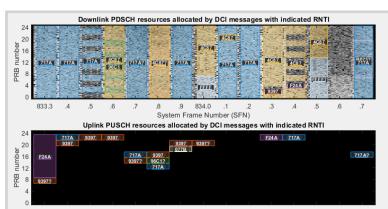
**Statistical Modeling of WLAN Interference on BLE Network**

## Signal Reception and Recovery

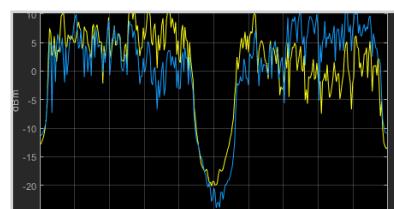
Wireless receivers must be designed to perform a range of information recovery operations to decode standard-based signals. These examples illustrate several aspects of receiver design for 5G, LTE, and WLAN systems.



**5G NR Synchronization Procedures**



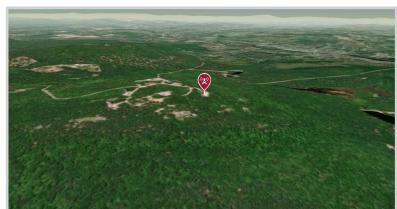
**LTE UE Detection Using Downlink Signals**



**802.11ax WLAN Signal Recovery**

## Channel Modeling and RF Propagation

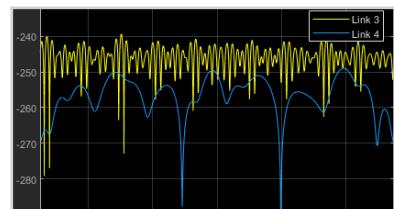
Wireless receivers need to mitigate impairments and propagation losses introduced by channel conditions. These examples illustrate a sampling of tools available in MATLAB toolboxes for modeling and visualizing the effect of various channels and RF propagation effects.



*Planning a 5G Fixed Wireless Access Link over Terrain*



*Visualize Antenna Coverage Map and Communication Links*

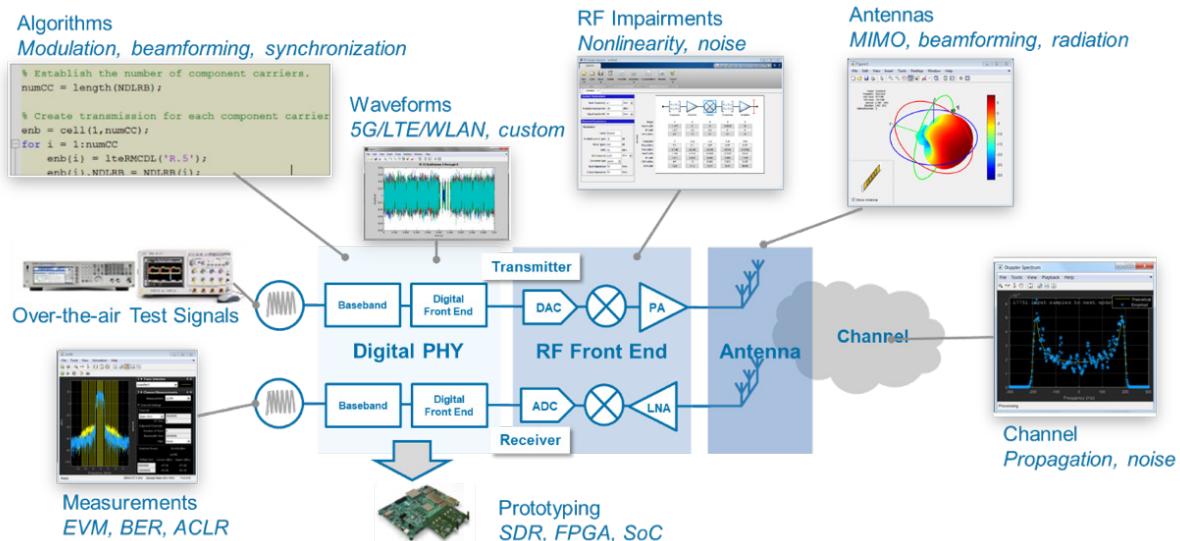


*Simultaneous Simulation of Multiple Fading Channels with WINNER II Channel Model*

## Joint RF-Baseband Modeling and Simulation

Traditionally, digital communications, RF, and antenna engineers have used different, isolated tools. Integration of components occurred when the first hardware prototype was developed. The advent of highly integrated RF transceivers and radio front-end technology has put a strain on this workflow. Now wireless engineers can use MATLAB and Simulink to model, simulate, and analyze complex RF front ends that include RF, analog, digital, and control logic components (Figure 7). The models are executable specifications of the RF front end that enable a workflow that encompasses:

- System-level simulation that is faster than circuit-level simulations
- Iterative development to keep models in sync with more detailed characterizations
- Validation of hybrid RF-digital designs before lab prototyping and testing



*Figure 7. Design, testing, and analysis of multidomain wireless systems with MATLAB.*

This workflow reduces time spent in validating and debugging system behavior, and improves collaboration with colleagues, suppliers, and customers. The following case studies show how the workflow can be applied to typical problems in high-speed, high-bandwidth communication systems.

### Case Study: Modeling Massive MIMO Antenna Arrays and Beamforming

MIMO antenna arrays effectively improve wireless system channel capacity, data rates, and interference rejection in LTE and Wi-Fi base stations and devices. New 5G mmWave designs will employ massive MIMO antenna arrays with hundreds of antenna elements on base stations (eNodeB). Because the area of an antenna array is reduced in proportion to the wavelength, an array for mmWave frequencies can be up to 100x smaller than an array for microwave frequencies.

Achieving an optimal design requires combined models of the antenna arrays and beamforming algorithms to simulate their interaction and impact on system performance. This puts a strain on current 3G and 4G design tools, which typically separate antenna design from system architecture and signal processing algorithms. MIMO simulation times are also typically 10x longer than 3G and 4G simulation times.

Behavioral-level simulation of the antenna array system reduces simulation time. This enables engineers to experiment with different array architectures and algorithms, simulate the performance of the array and associated algorithms, adjust parameters to mitigate the effect of antenna coupling, and assess tradeoffs between antenna gain and channel capacity. Assessing these tradeoffs can be done iteratively with Phased Array System Toolbox™ and Antenna Toolbox™ for MIMO array design tasks.

Figure 8 shows a beam pattern and grating lobe diagram for 66 GHz 64x64 element design, designed with Antenna Toolbox.

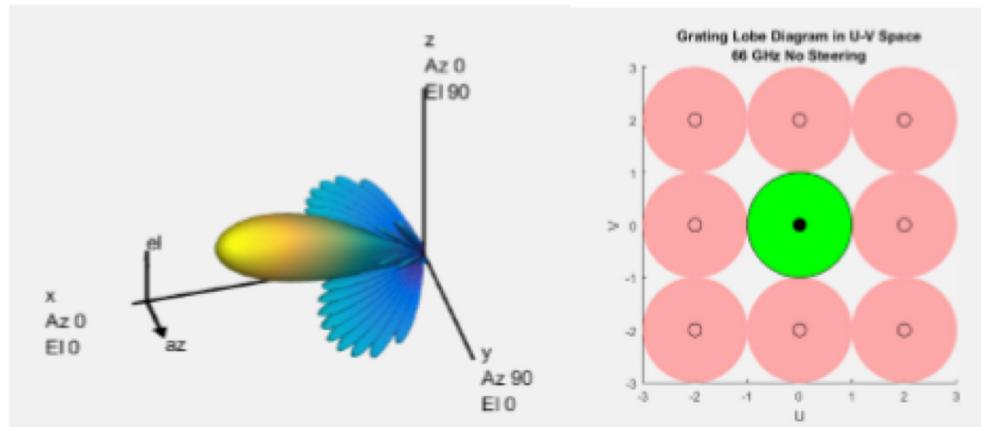


Figure 8. Beam pattern and grating lobe diagram for 66 Ghz 64x64 element design.

## Hybrid Beamforming

Beamforming can mitigate the signal path and propagation challenges at mmWave frequencies. It would be ideal to have independent weighting control by dedicating a transmit/receive (T/R) module to each antenna element, but this is generally not practical due to cost, space, and power limitations.

Hybrid beamforming is a technique to partition beamforming between the digital and RF domains to reduce the cost associated with the number of RF signal chains. Hybrid beamforming combines multiple array elements into subarray modules, with one T/R module dedicated to a subarray in the array.

To help meet the required performance parameters within the implementation cost constraints, Simulink enables unified, multidomain modeling and simulation of the RF domain and digital domain components. Circuit envelope simulation in RF Blockset™ ensures fast simulation of the hybrid system.

In Simulink, you can evaluate the number of elements and precoding parameter values to ensure system-level performance is met across a range of steering angles. The digital and RF beamforming weights can be calculated in MATLAB and incorporated into the Simulink model.

Figure 9 shows a section of a multidomain model containing digital beamforming weights used to shape the signals feeding the RF subarrays, where phase shifts are applied. The resulting hybrid weights produce the desired array pattern.

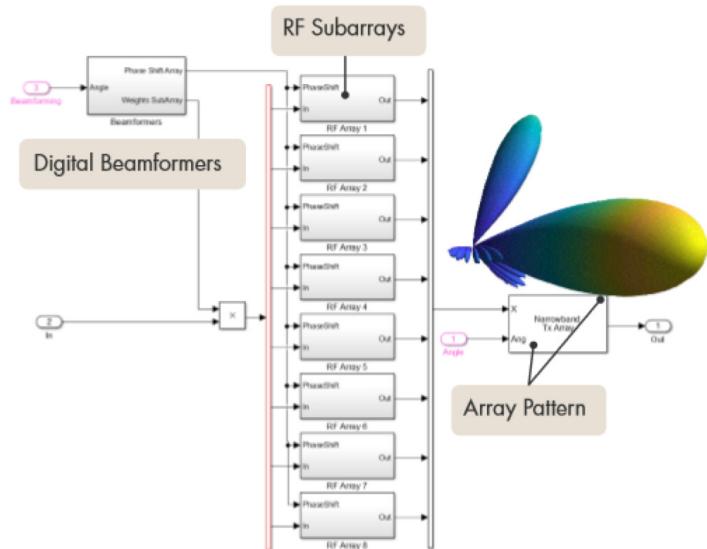
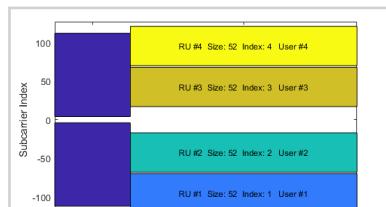


Figure 9. Hybrid beamforming design in MATLAB.

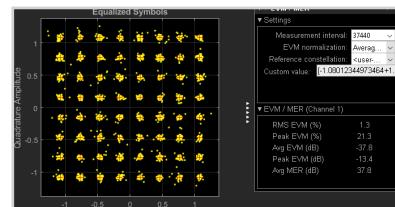
## Learn More About MIMO Processing and Beamforming in Wireless Systems



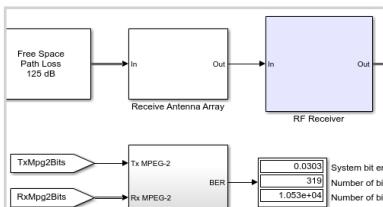
*Hybrid Beamforming for Massive MIMO Phased Array Systems*



*802.11ax Downlink OFDMA and MU-MIMO Throughput Simulation*



*MIMO-OFDM Precoding with Phased Arrays*



*Digital Video Broadcasting with RF Beamforming*

### Case Study: Power Amplifier Characterization with DPD for Reduced Signal Distortion

The high efficiency required of power amplifiers (PAs) to operate at higher frequencies and larger bandwidths introduces nonlinearities. Digital predistortion (DPD) techniques can increase the efficiency of the transmitter while limiting spectral regrowth and inter-channel interference.

Developing a quality DPD algorithm is challenging, because it requires an understanding of the effects introduced by the power amplifier. Because power amplifiers are nonlinear and are affected by finite memory, the characterization of PAs strongly depends on the bandwidth of the driving signal, its spectral occupation, and its peak-to-average-power (PAPR) ratio.

Because of this complexity, DPD algorithms are often developed in the lab, using prototyping hardware to test algorithms together with the actual PA. While this approach is useful to validate algorithms, it is hard to explore the DPD design space, and cannot be used when the actual PA is not yet available.

Using behavioral models enables development of DPD algorithms before lab prototyping. With RF Blockset, you can identify a high-quality memory polynomial models using PA input/output characteristics coming from circuit-level simulation tools or from lab measurements. In a matter of seconds, you can explore different polynomial orders and memory depths (Figure 10).

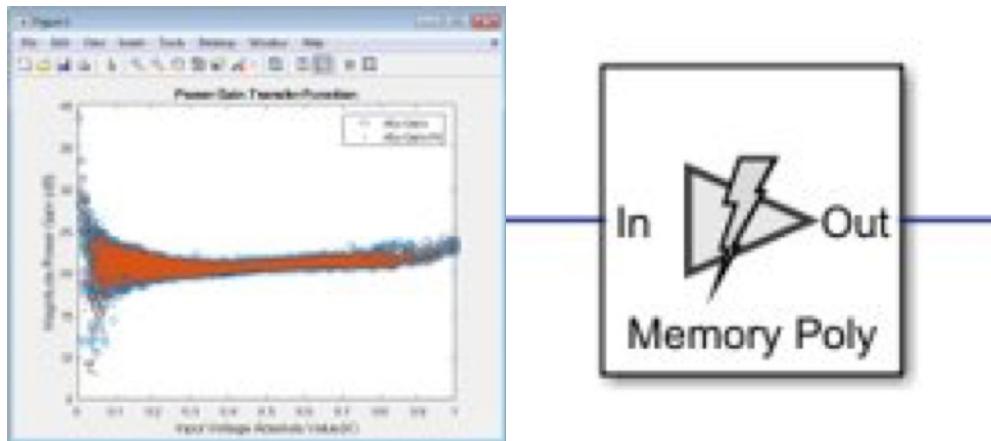


Figure 10. Nonlinear PA model with memory effects derived from measured power amplifier data in MATLAB (left), and the corresponding Simulink PA model for use in time-domain circuit envelope simulation (right).

The PA model can be used within a system simulation together with realistic (and standard-compliant) baseband signals, models for the RF transmitter and observer receiver, the antenna termination expressed with S-parameters, and various adaptive DPD algorithms (Figure 11). With this approach, designers can rapidly validate designs while accounting for dispersive and nonlinear effects that are difficult to capture in the lab.

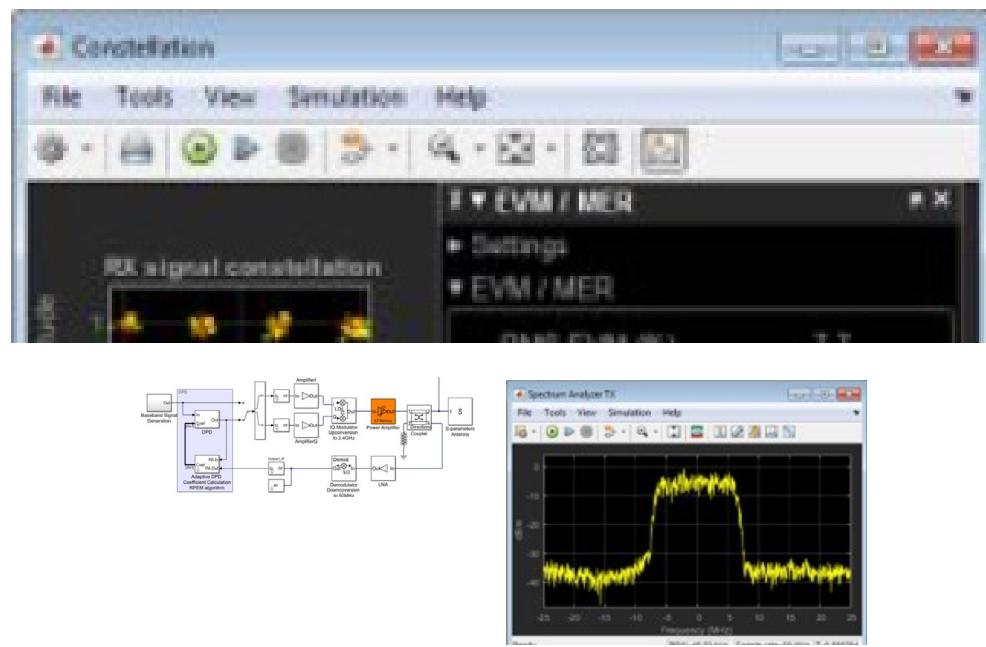


Figure 11. Closed-loop simulation of an RF power amplifier with adaptive DPD (top) and resulting output spectrum and EVM measurements (bottom).

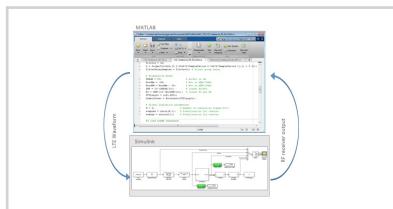
## Learn More About Joint RF-Digital Modeling and Simulation



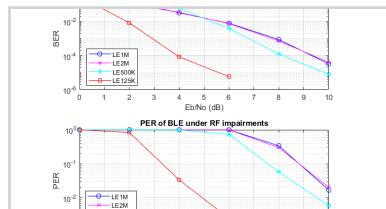
*Four Steps to Building Smarter  
RF Systems with MATLAB*



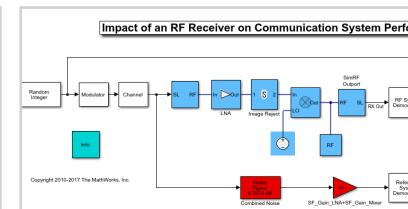
*Model RF Power Amplifiers and Increase  
Transmitter Linearity with DPD Using MATLAB*



*RF Receiver Modeling  
for LTE Reception*



*Bluetooth Low Energy PHY  
Simulation with RF Impairments*



*Impact of an RF Receiver on  
Communications System  
Performance*

## Design Verification and Testing

### Standard-Compliant Signal Generation and Measurement

Testing is too important to be left until you get to the hardware lab. As soon as you start developing an algorithm or designing an RF front end, you need a test stimulus and measurements to assess performance and debug problems.

Requirements at the simulation stage can be quite different from those during the testing phase. Researchers and design engineers need greater flexibility to customize waveforms, visualize data and access low-level information to find subtle problems, and run simulations over a wide range of scenarios.

Using MATLAB, you can:

- Generate a standard-compliant signal (Figure 12) interactively or with a few lines of MATLAB code
- Add impairments to waveforms
- Analyze test results with scopes and measurements such as EVM and ACLR
- Verify designs with simulations or live over-the-air tests
- Deploy test applications to C/C++ testing environments

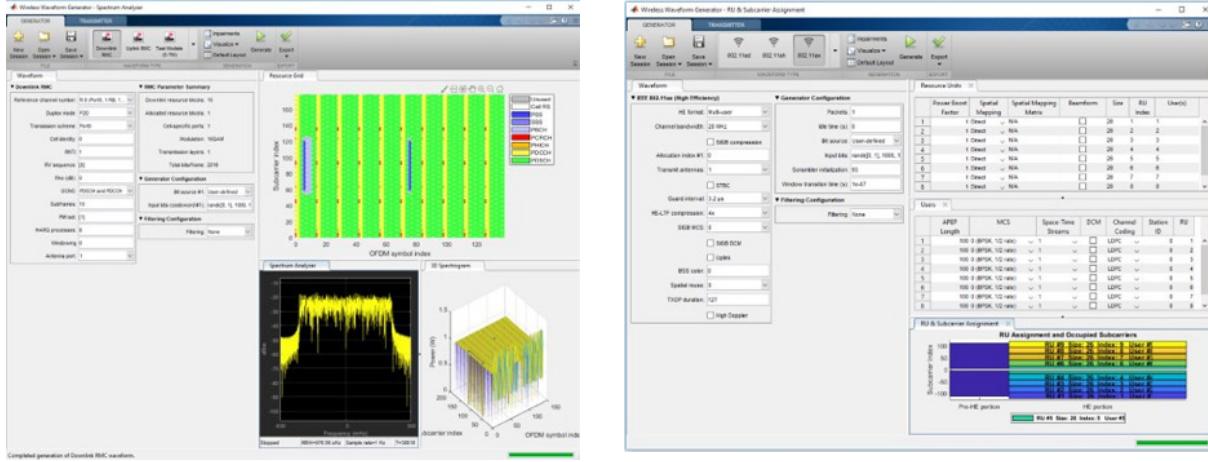
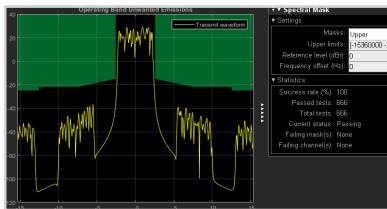
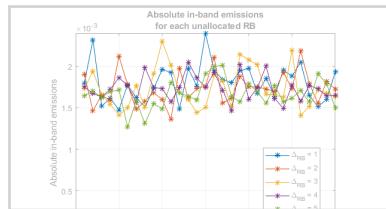


Figure 12. Using Wireless Waveform Generator App to generate standard-based test waveforms: LTE (left) and 802.11ax (right).

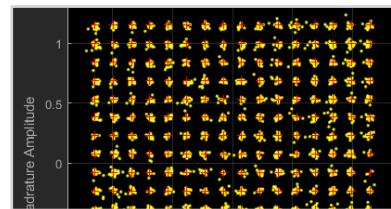
## Learn More About Standard-Based Test and Measurement with MATLAB Toolboxes



LTE Downlink ACLR Measurement



LTE Uplink EVM and In-Band Emissions Measurements



802.11ac Transmitter Modulation Accuracy and Spectral Emission Testing

## Over-the-Air Testing with RF Instruments and Software-Defined Radios

By connecting to software-defined radio and RF instrument hardware, you can perform over-the-air tests of your MATLAB code and Simulink models at your desktop, in the lab, or in the field to validate designs under real-world conditions. The test benches, signal generators, scopes, and measurements used at the simulation stage can be reused for hardware testing. This approach eliminates the need to recreate tests in a different software environment, reducing test development time and errors.

The flexibility and analysis capability of MATLAB enable you to diagnose and debug subtle issues that are difficult to address with the constrained software environments typically associated with test and measurement hardware.

MATLAB and Simulink support a range of available SDR hardware as well RF signal generators and spectrum analyzers from Keysight, Rohde & Schwarz, National Instruments, Anritsu, and other RF instrument vendors. You can choose the hardware that's most appropriate for your requirements and budget (Figure 13).

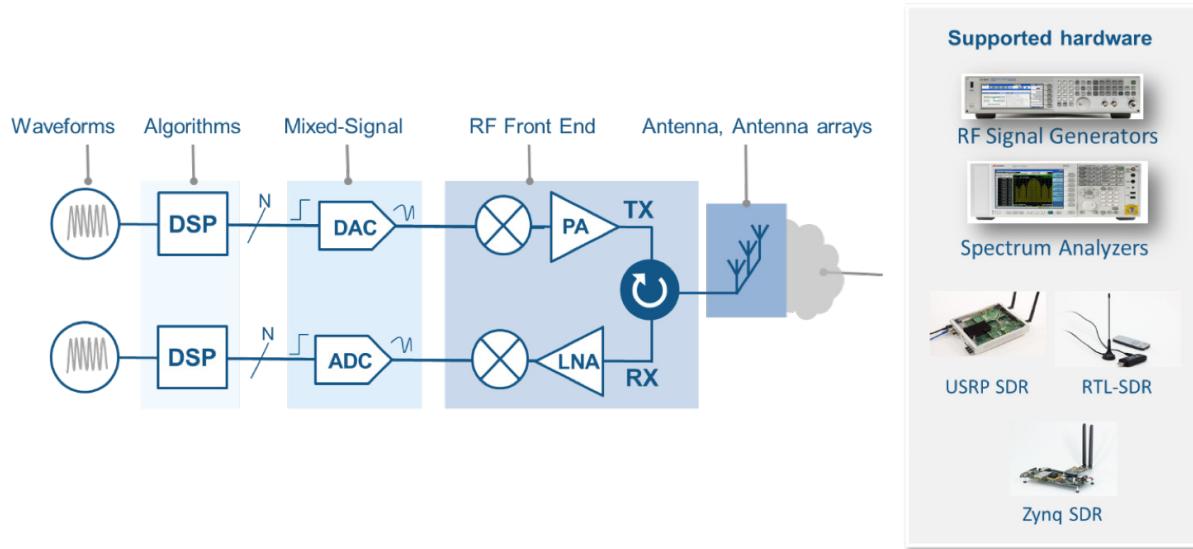


Figure 13. MATLAB and Simulink support for over-the-air testing with a range of available SDR and RF instrument hardware.

### Case Study: LTE Scanner with SDR Hardware

This example uses MATLAB and LTE Toolbox™ with an SDR platform to scan a frequency band for an available LTE signal from nearby base stations, as depicted in Figure 14, and decode cell-specific information.

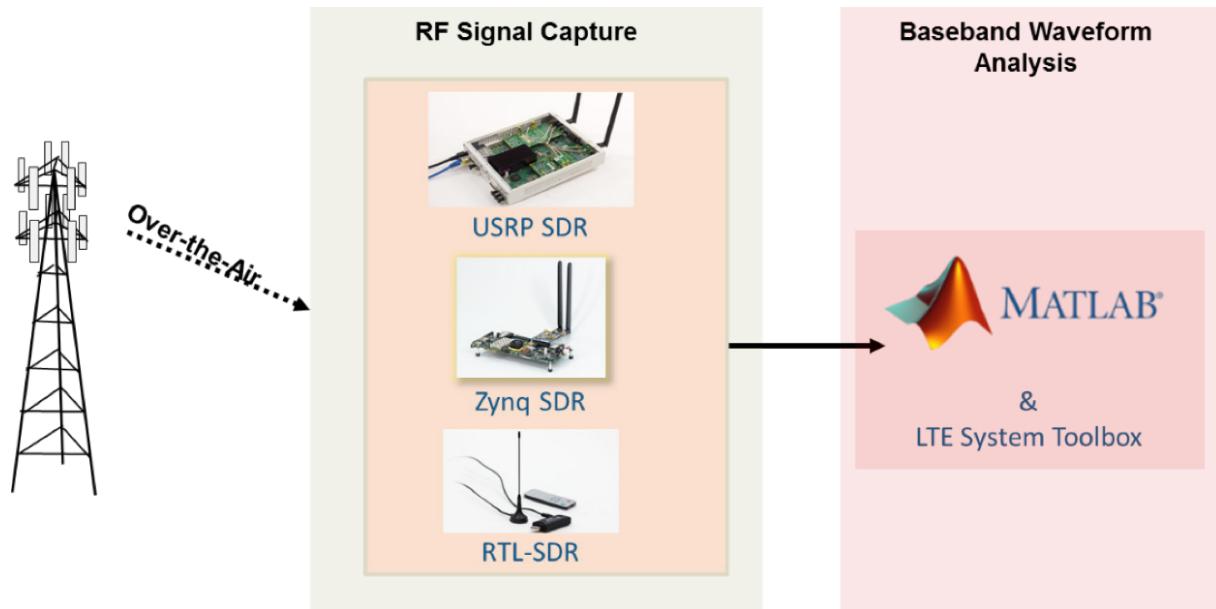


Figure 14. Using SDR hardware with MATLAB to scan for an LTE signal. The analysis is hardware-independent and can be used with any of the depicted radio platforms.

The SDR platform searches the whole band (or portions of it) using carrier frequencies in steps of 100 kHz. The captured signal is demodulated to baseband and passed to LTE Toolbox to perform the cell search procedure. The scanner can decode the master information block (MIB) and system information block 1 (SIB 1) to provide information about the found cells.

The app shown in Figure 15 drives the signal capture and decoding operations. It can also plot the spectrum of the whole selected band to help identify where the LTE signals may be located. The app provides simple control over key parameters of the algorithm, and finer control or different information may be obtained by adapting the corresponding MATLAB code.

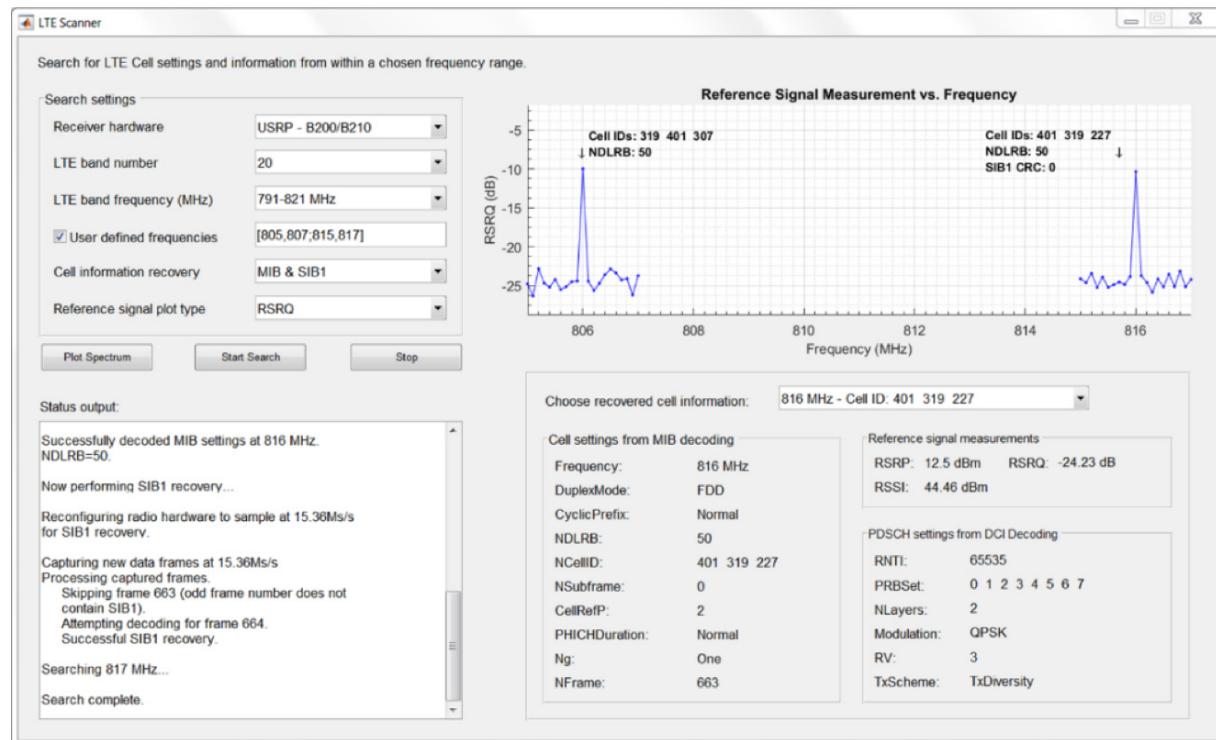
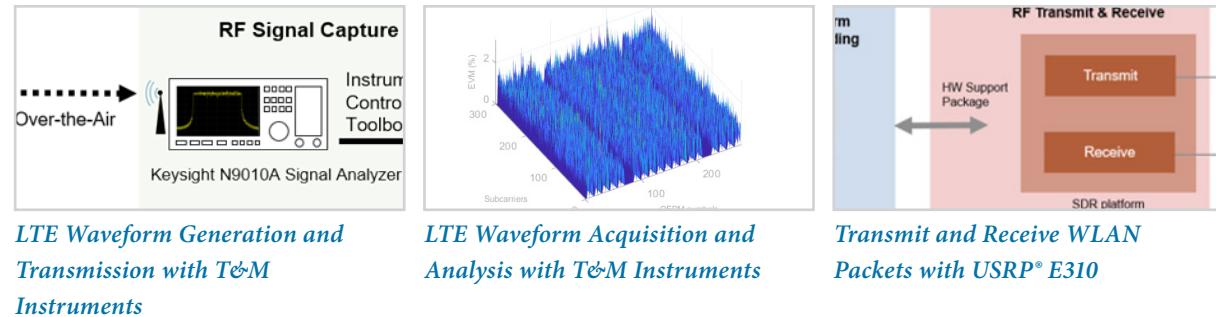


Figure 15. LTE cell scanner app.

## Learn More About Testing with RF Instruments and Software-Defined Radios



## Wireless Prototyping and Implementation

### Prototyping Workflow for FPGA and SoC-Based SDR Platforms

You can employ Model-Based Design to design a wireless transmitter or receiver model and generate HDL and C code onto FPGAs and SoCs. Once you have a floating-point design running in MATLAB that has been validated with simulation and over-the-air tests, you can iteratively convert algorithms to hardware-ready fixed-point Simulink models and deploy the design to the target system (Figure 16).

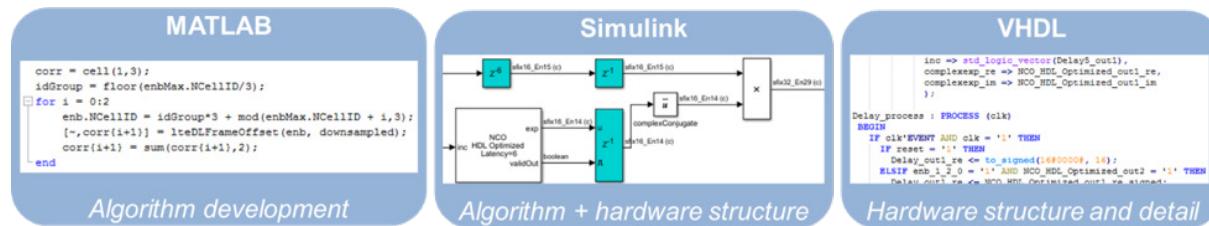


Figure 16. Algorithm and hardware design environments. Algorithm developers use MATLAB; hardware designers use VHDL or Verilog. Simulink bridges this gap, enabling these two roles to collaborate toward a higher-quality implementation.

As you prototype each element of the design, the rest of the model on your PC serves as a test bench for your target implementation. The workflow supports partitioning of the design to generate HDL and C code for the hardware and software portions of implementation. Figure 17 shows the incremental process from pure simulation to RF testing.

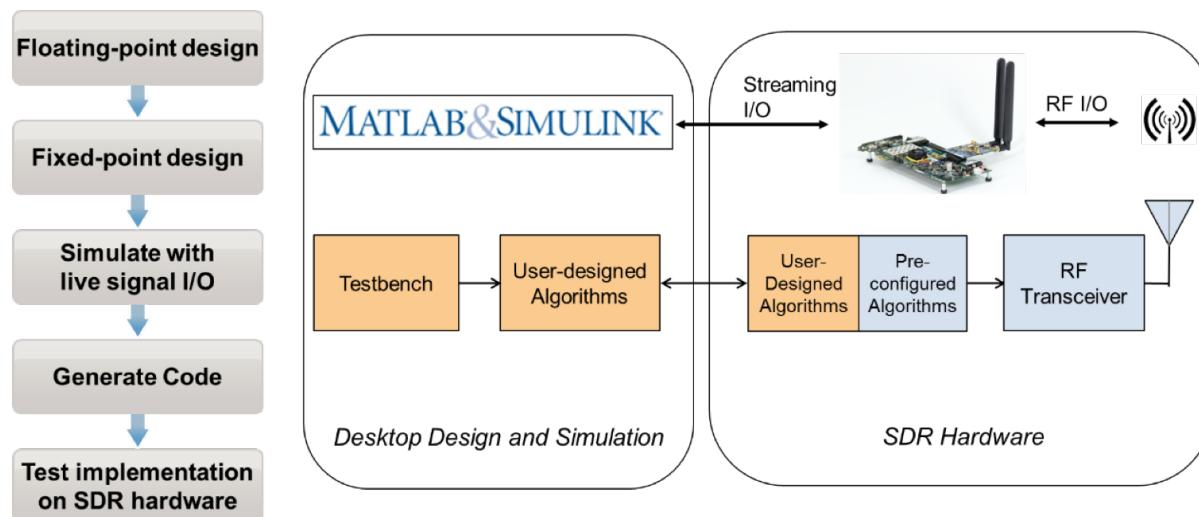


Figure 17. Workflow for migration of algorithms from simulation to hardware. User-designed components are shown in orange, and the preconfigured SDR framework is shown in blue.

The steps in the workflow include:

- Partitioning the Simulink model into functions that will target the FPGA fabric and the ARM® processor on a programmable SoC
- Refining the model to optimize the performance of the generated HDL code
- Generating HDL and C code from the algorithm models
- Integrating and testing the generated code on the radio platform

At the end of this process, a fully verified SDR system will be produced, running C and HDL code automatically generated from a Simulink model and transmitting or receiving signals in real time.

### **Case Study: Optimized HDL Code for FPGA and ASIC Implementation**

Model-Based Design doesn't stop at prototyping. Wireless engineers successfully use MATLAB and Simulink with automatic HDL generation to produce algorithm implementations that meet the performance, size, and power requirements of production FPGA and ASIC designs. The iterative workflow enables rapid development and verification of highly efficient hardware implementation of algorithms such as multirate filtering, digital predistortion, and baseband processing.

Figure 18 shows a Simulink model of an LTE OFDM modulator and detector (top) and an HDL implementation-ready model of the detection algorithm (bottom). The detector performs frequency estimation and correction, PSS detection, timing adjustment, FFT, and SSS detection in order to determine the LTE cell identity of a detected cell group. The model was verified using an LTE-compliant signal generator and successfully detects cell IDs both in simulation and over-the-air running on a Zynq® SDR. The model produces hardware-independent HDL code that can be used on any FPGA or ASIC. Optimizations for other architectures can be performed by modifying the model and regenerating the HDL code and verified using the Simulink test bench.

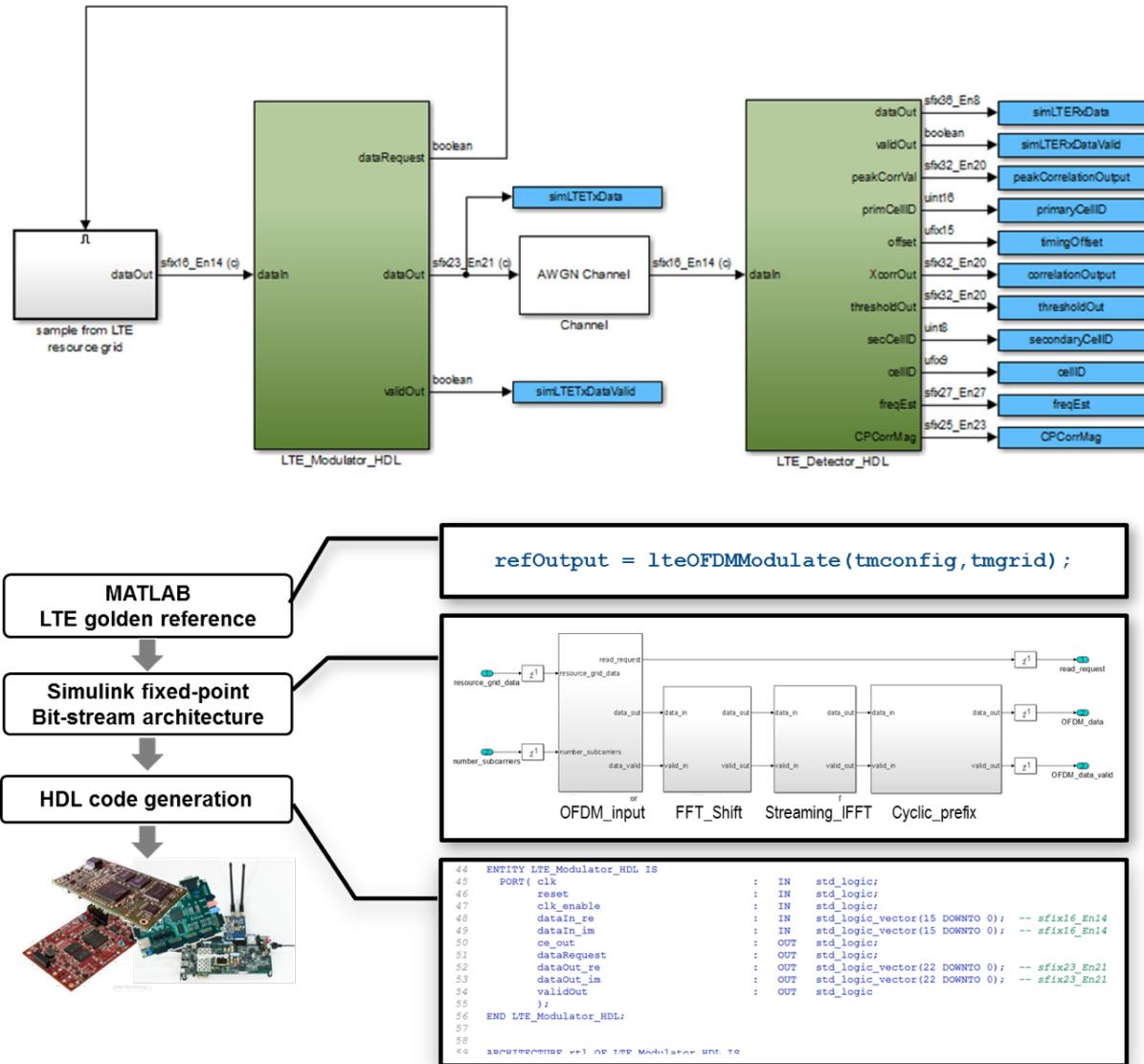


Figure 18. Simulink model for an LTE-compliant modulator and detector (top), and workflow for designing and verifying an HDL implementation of the modulator subsystem (bottom).

## Learn More About Wireless Prototyping and Implementation



*Deploying LTE Wireless Communications on FPGAs*

## Summary

The race to keep pace with advances in wireless technology is driving innovation in tools and methods for wireless design. Advanced wireless products require an ever-increasing level of technology integration.

Successful wireless engineering teams recognize that they need to work in an environment that can encompass algorithm design, system simulation, over-the-air testing, prototyping, and implementation. Compared with groups still designing in silos, teams taking advantage of earlier design integration available with MATLAB and Model-Based Design report saving as much as 30% in overall development time and 85% in functional verification, having dramatically fewer design respins, and creating defect-free implementations on the first attempt.

Many of those teams recognize that they can deliver working products faster by simulating, prototyping, and verifying their designs using MATLAB. To learn more and request a free trial, visit [mathworks.com/wireless](http://mathworks.com/wireless).

## Learn More About Wireless Design with MATLAB

- [MATLAB for Wireless Communications](#) - Overview

## See How Leading Companies Are Using MATLAB for Wireless Design

- [Qualcomm Uses MATLAB to Develop 5G RF](#) - Case Study
- [InterDigital: Advancing the 5G Wireless Standard](#) - Case Study
- [Huawei Develops a Radio Frequency System](#) - Case Study
- [Nokia: Model-Based Design for 5G](#) - Case Study
- [NanoSemi Improves System Efficiency for 5G and Other RF Products](#) - User Story
- [BT Optimizes System Performance of Femtocell Deployment](#) (2:36) - Video
- [DigitalGlobe Simulates Complete Satellite-to-Ground Communications Systems](#) - User Story
- [Ericsson: Radio Testbed Design Using HDL Coder](#) (22:44) - Video