

Transforming Wireless Design with MATLAB

```
n(m,9); end
in(n,2); end

rst 12 membrane

5, 15.19725192, 2*pi
488, 5*pi^2, 5*pi^2

1 2 1];

lar coordinates

; ones(2*m,1)./sin(
17 19 ... ] * 2/3,
16 20 ... ] * 2/3,
21 ... ] * 2/3, (mu

));
-a));
```



Contents

The Expertise Explosion.....	3
Key Requirements for Today’s Wireless R&D.....	4
Integrated Environment for RF-Digital Modeling and Simulation.....	4
Working with Standard-Compliant Systems and Signals.....	5
Over-the-Air Testing with Software-Defined Radios and RF Instruments.....	6
Accelerating Hardware and Software Development with Model-Based Design.....	7
Wireless System Design with MATLAB.....	9
LTE, Wireless LAN, and the Road to 5G.....	10
Case Study: LTE 8x8 Throughput Simulation.....	11
Case Study: LTE Framework for Evaluating Candidate 5G Modulation Schemes.....	12
Case Study: Massive MIMO for Interference Reduction.....	14
Joint RF-Baseband Modeling and Simulation.....	16
Case Study: Modeling a Tunable RF Transceiver.....	16
Design Verification and Testing.....	18
Standard-Compliant Signal Generation and Measurement.....	18
Over-the-Air Testing.....	20
Case Study: LTE Scanner.....	21
Model-Based Design for 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
Summary.....	24

Innovation in wireless communications is stretching the old ways of doing development to their limits. That's because the latest innovations require a new level of technology integration.

Successful wireless engineering teams recognize that they need a software 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 report saving as much as 30% in overall development time and 85% in functional verification time, having dramatically fewer design re-spins, and creating defect-free implementations on the first attempt.

In this ebook, we examine key challenges in the changing landscape of wireless system design that affect the speed and success of product development. In the process, we introduce a more efficient way of doing wireless system research and design. Areas we examine include:

- Component integration and simulation during the early stages of R&D
- Modeling and testing approaches to ensure compliance with standards at each design stage
- Software to exploit popular software-defined radio (SDR) hardware for over-the-air testing and prototyping
- Model-Based Design as a way to shrink design cycles

The Expertise Explosion

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.

When working on 5G or other advanced technologies, wireless engineers have to focus on accelerating research and innovation, and minimize the time and resources spent on programming, debugging software, or building FPGA hardware realizations of their design ideas. For a typical R&D group that consists of engineers with strong signal processing and algorithm development backgrounds but relatively little experience with RF design or hardware implementation, it is often difficult to simulate or prototype complete radio designs without outside assistance. They are similarly challenged by problems such as developing massive multi-antenna systems that depend on understanding the behavior of highly coupled digital, RF, and antenna elements.

The expertise needed to develop wireless technology is exploding. This is major challenge for R&D teams as they race to deliver the next breakthrough. The days of domain experts working separately, using separate tools, are numbered. Fortunately, there is a better way.

Key Requirements for Today's Wireless R&D

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

- Design algorithms and RF components in the context of an end-to-end simulation
- Create standard-compliant waveforms and reference models to verify designs
- Use cost-effective SDR hardware to test designs with live over-the-air signals
- Develop and verify embedded radio prototypes and implementations

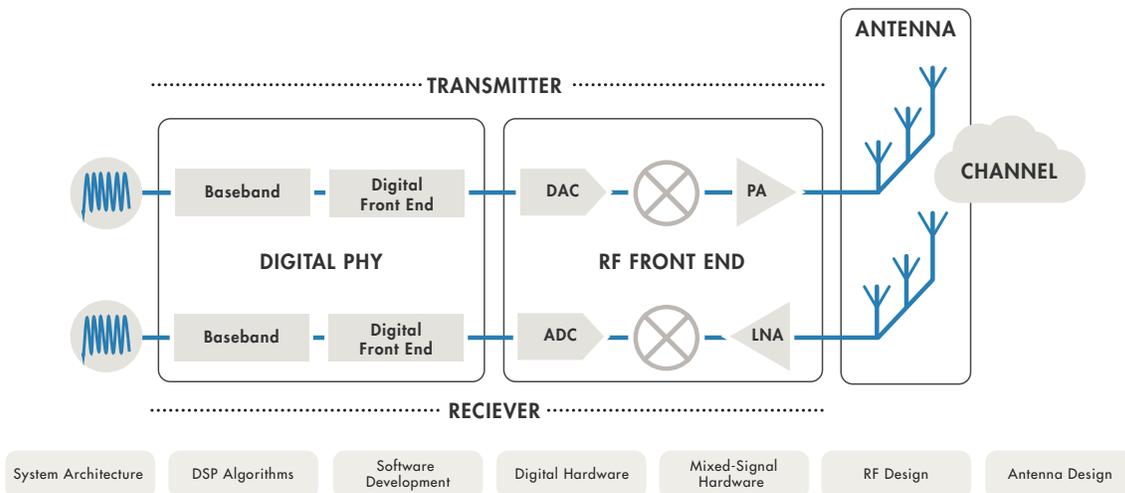


Figure 1. The many engineering skills and workflows required for efficient development of modern wireless systems.

Integrated Environment for RF-Digital Modeling and Simulation

Higher data rates, massive connectivity, and 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. In this environment, engineers can design, model, and simulate multidomain wireless systems. Domain experts in each area can use the tool that is best suited for their problem, to model RF architectures, digital hardware, and complex state machine logic.

This environment can include a full model of each component within the wireless system as well as channel and interference models. 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 research and design problems such as:

- Modeling multi-antenna (MIMO) systems found in LTE and WLAN systems and 5G proposals, including antenna arrays, propagation patterns, and beamforming
- Simulating end-to-end communication links, including RF and channel impairments and digital algorithms to mitigate those impairments
- Modeling digitally controlled “smart” RF transceivers, providing fast, accurate simulation results and control over hundreds of parameters that are difficult to tune and debug in the hardware
- Exploring, designing, and analyzing algorithms such as frequency, time, and carrier recovery and advanced modulation to improve throughput, efficiency, and robustness

Working with Standard-Compliant Systems and Signals

Developing and testing products for the LTE and WLAN wireless standards requires tools to ensure compliance with those standards. Those tools must meet different requirements at each stage of development, including research, design, and testing.

Researchers need software that can be modified easily to explore alternative algorithm designs. The in-house simulation tools often developed for this task are poorly documented and difficult to adapt for other purposes, such as prototyping and testing. For those tasks, design and test engineers may use software supplied with test and measurement equipment. This software provides a necessary source for testing, but cannot be used independently of the test hardware or modified 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"> Assess how design choices affect system performance Confirm that the system complies with the standard
Golden reference for design verification	<ul style="list-style-type: none"> Verify that the design works as it should
Standard-compliant signal generation	<ul style="list-style-type: none"> Test the design with synthetic or live over-the-air signals
Signal detection, analysis, and information retrieval	<ul style="list-style-type: none"> Decode real-world signals Validate receiver design

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

In order to span this set of tasks, the software must 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
- An interactive, high-level programming language to facilitate experimentation, detailed modeling, 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

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

While simulation is essential, ultimately a wireless system must transmit and receive signals over the air. With the growing commercial availability of inexpensive 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, making reuse impossible. With recent advances, it is now practical to connect popular 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.

The ideal software is hardware-agnostic, so engineers can:

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

Accelerating Hardware and Software Development with Model-Based Design

Wireless engineers are under increasing pressure to deliver first-to-market technology. To this end, they increasingly rely on early hardware prototypes and proof-of-concept testbeds using FPGA or programmable SoC hardware. While powerful commercial prototyping platforms are now available, these engineers frequently lack the HDL expertise to create those prototypes.

When working on fast-changing wireless technologies, system engineers and architects must focus on accelerating research and innovation, and minimize the engineering time and resources spent on programming, debugging software, or building FPGA hardware or SoC hardware-software realizations of their design ideas. The tools and workflows that interface to prototypes and testbeds must support rapid design iterations and rapid deployment of new algorithms or design changes.

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

- Create an executable specification for system behavior
- Develop and verify fixed-point hardware-ready designs in simulation
- Automatically generate HDL and C code
- Automate the workflow for prototyping on SDRs and other FPGA or SoC hardware
- Reuse the generated code and test benches for production implementation and verification

With Model-Based Design (Figure 2), models serve as executable specifications that enhance understanding of complex designs and collaboration among engineers, automatic code generation from models reduces implementation errors, and continuous verification significantly decreases 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.

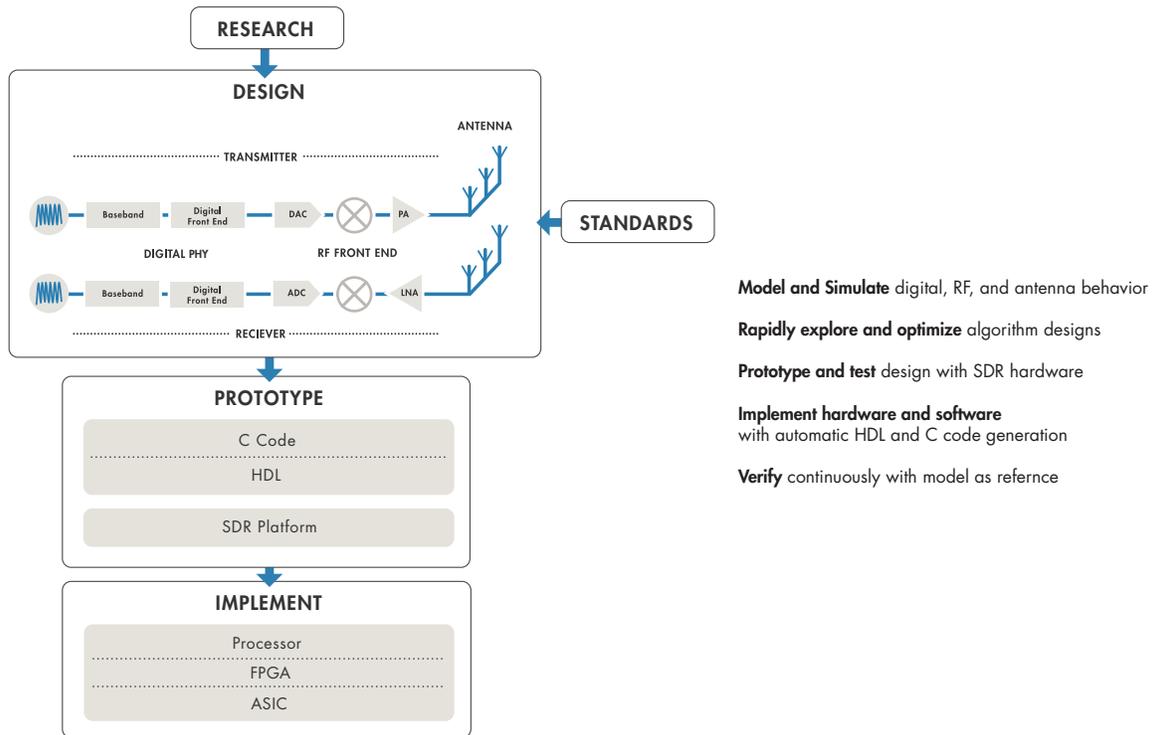


Figure 2. Unifying and accelerating the wireless development workflow by using Model-Based Design.

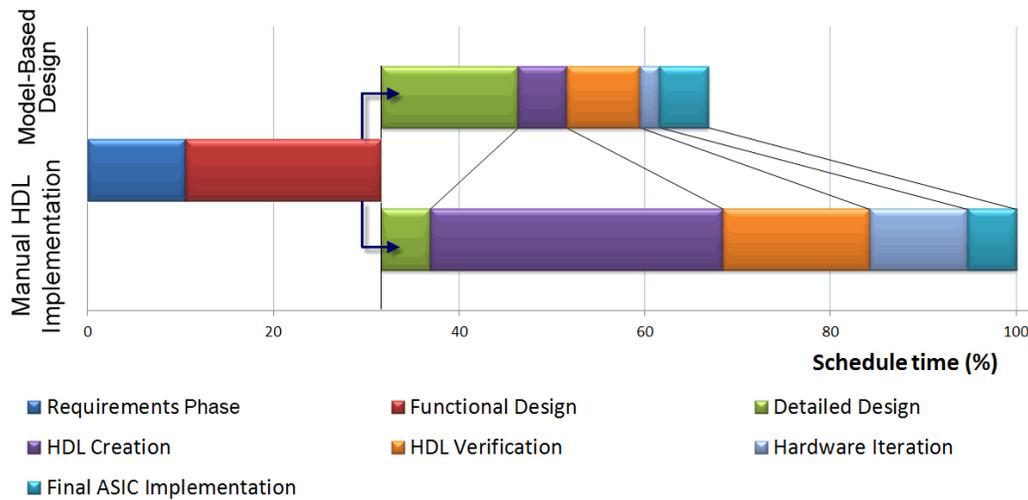


Figure 3. Time savings using Model-Based Design to develop HDL IP for wireless ASICs.

Wireless System Design with MATLAB

Many wireless engineers rely on MATLAB® to develop algorithms, analyze data, explore new technologies, and publish thousands of research papers and contributions to the MATLAB Central community site. 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).

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 verification with MATLAB and Simulink reference models
- 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.

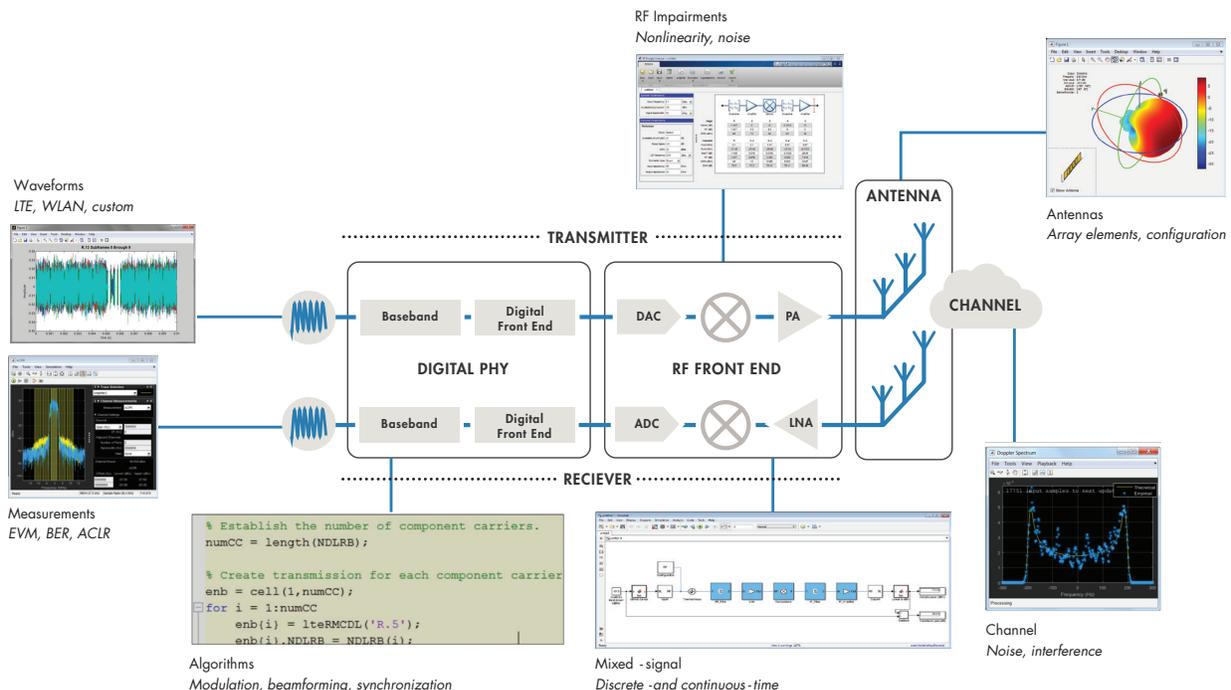


Figure 4. Design, testing, and analysis of multidomain wireless systems with MATLAB.

LTE, Wireless LAN, and the Road to 5G

As the wireless industry turns to 5G to provide ultrafast, ubiquitous mobile access, leading technology experts predict that LTE and WLAN will continue to evolve and play an important role in future systems (Figure 5). They will be tightly integrated with new radio access technologies to form the new 5G networks. Researchers are investigating a number of technology areas to achieve this goal, including:

- LTE and WLAN evolution
- New modulation schemes: FBMC, UFMC, GFDM, and others
- Massive MIMO antenna arrays and beamforming
- New frequency bands: mmWave
- Advanced receivers

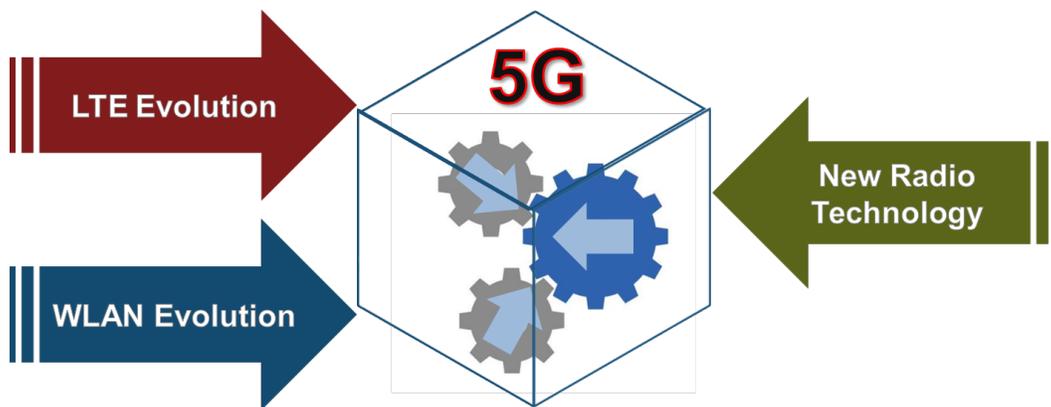


Figure 5. 5G systems integrating new radio technologies with evolving LTE and WLAN standards.

Investigating and proving viable strategies for these 5G technologies involves techniques that cross the traditional disciplines of RF, antenna, and baseband design. R&D teams can rapidly evaluate candidate 5G technologies right in MATLAB with:

- Integrated antenna models, phased array design, and baseband processing
- Existing standard-compliant reference models that can be modified to evaluate the performance improvement of new proposed techniques
- Scopes and measurements to analyze and visualize simulation results

The next three examples show case studies of state-of-the-art standard models, modifications of existing systems to study the performance of proposed 5G modulation techniques, and the investigation of massive MIMO for reducing interference.

Case Study: LTE 8x8 Throughput Simulation

LTE, WLAN, and proposed 5G wireless systems use multi-antenna (MIMO) configurations to improve throughput and interference rejection. These systems employ sophisticated signal processing to control the transmission and reception of MIMO signals. A number of schemes have been employed to accommodate different scenarios. MATLAB supports simulation of MIMO systems with LTE-compliant or WLAN-compliant waveforms, phased antenna array models, and the physical layer and MIMO processing algorithms.

For example, the LTE standard supports multi-antenna (MIMO) configurations to improve throughput and/or signal strength. Within the standard, beamsteering, diversity, or spatial multiplexing techniques can be used.

In this example, we simulate a range of scenarios that can accommodate up to an 8x8 antenna configuration as specified in Release 12 of the LTE standard, illustrated in Figure 6. The simulation supports TDD duplexing and either beamforming or spatial multiplexing.

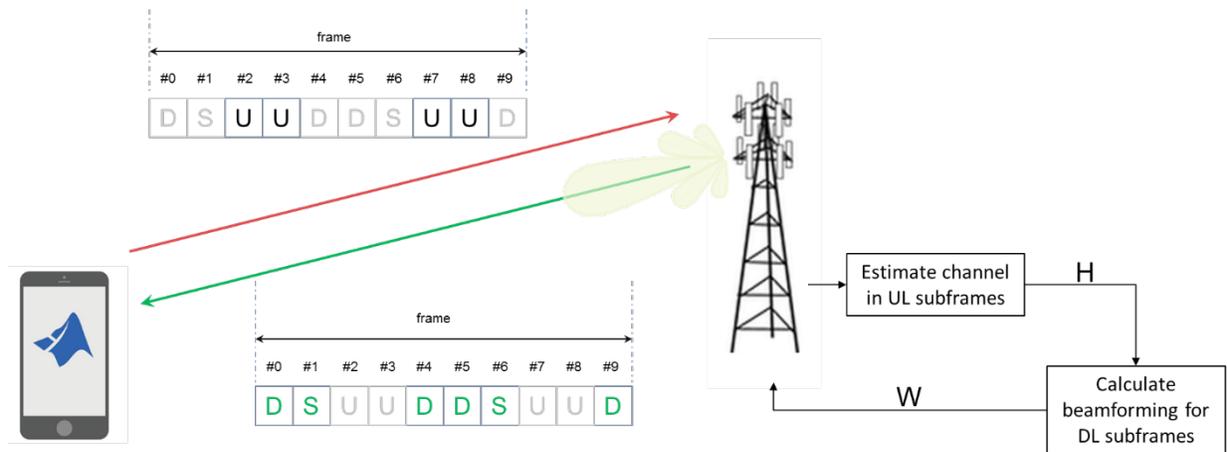


Figure 6. LTE throughput simulation scenario using an 8x8 antenna configuration and beamsteering.

The TDD scenario models transmission modes (TMs) TM7, TM8, TM9, and TM10. It exploits channel reciprocity and performs channel estimation in the UL subframes. Using this channel estimate, we calculate the beamforming matrix to be used in the downlink subframes. This process is illustrated in Figure 7.

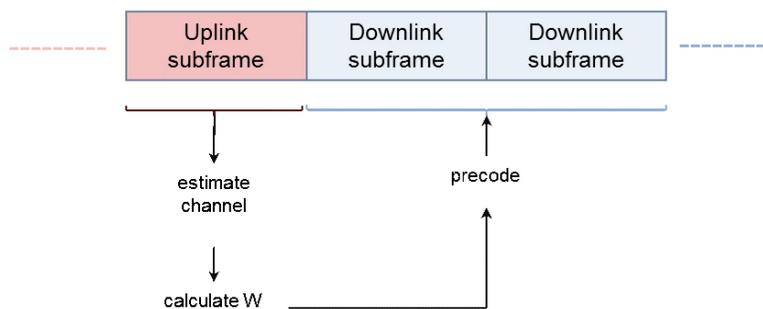


Figure 7. Calculation of beamsteering matrix from uplink channel estimate.

By modifying the number of layers and the number of transmit and receive antennas, we can calculate the throughput curves for given channel conditions, as shown in Figure 8. This can help determine the most appropriate transmission mode for a given set of propagation conditions.

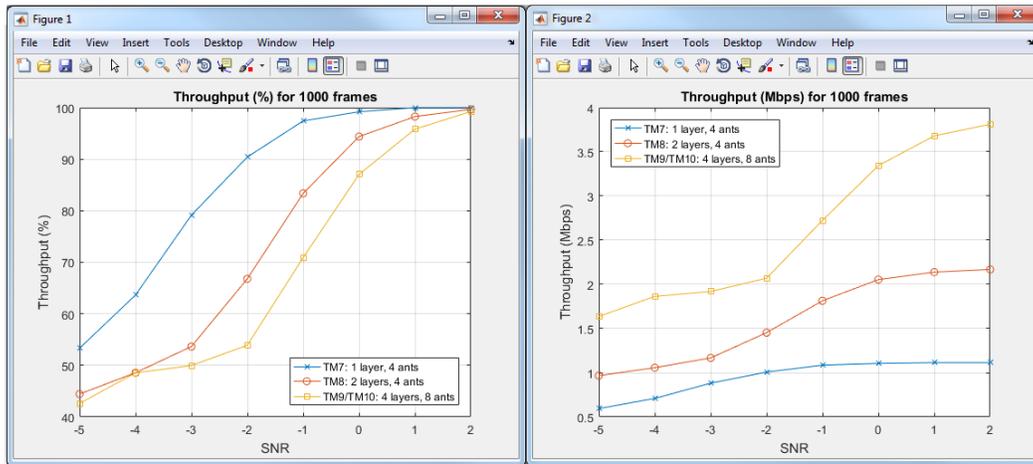


Figure 8. Throughput curves for different number of layers and antennas.

Case Study: LTE Framework for Evaluating Candidate 5G Modulation Schemes

New modulation schemes have been proposed for 5G systems. These schemes have common characteristics to meet 5G requirements, including improved spectral efficiency and relaxed synchronization requirements. As a result, the majority of candidates involve multicarrier, non-orthogonal waveforms. Some candidates are:

- OFDM (the incumbent used in LTE and WLAN)
- FBMC: Filter-Bank Multi-Carrier
- UFMC: Universal Filtered Multi-Carrier
- GFDM: Generalized Frequency Division Multiplexing

Determining which scheme will best meet 5G requirements involves studying performance and efficiency tradeoffs. Consider the differences between UFMC, in which filtering is applied per subband, and FBMC, in which filter is applied per subcarrier, as shown in Figure 9.

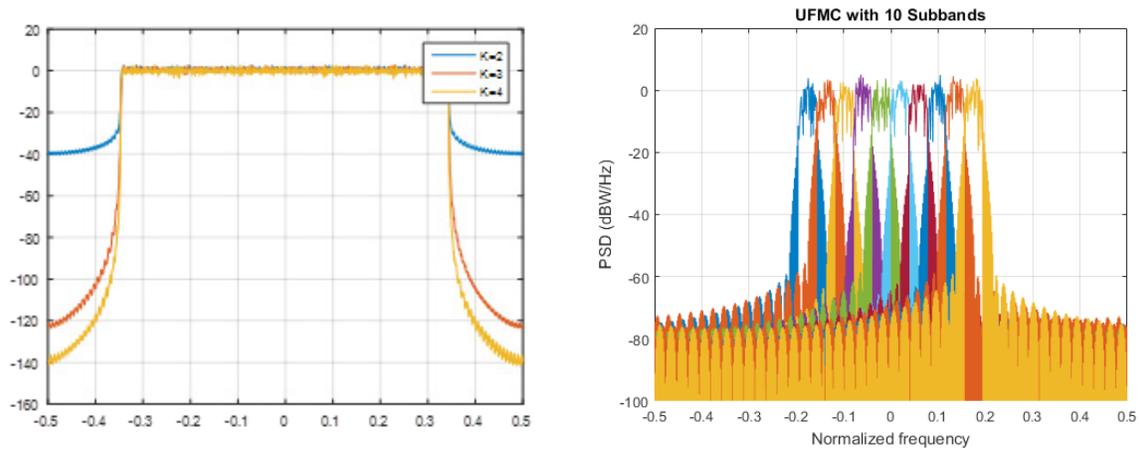


Figure 9. The spectral characteristics of FBMC (left) and UFMC (right).

The proposed modulation schemes mentioned above are just a small part of the system, and their performance cannot be fully studied in isolation. They need to be part of a larger system. For example, to calculate the throughput of a link using FBMC or UFMC, we need to perform operations such as channel estimation and equalization. Reference signals are required for channel estimation. As a first approach we can use the signal structure of existing standards, like LTE.

Figure 10 shows how we can generate an LTE resource grid, including the reference signals. This can be fed to a UFMC modulator, sent through a channel, and equalized and decoded at the receiver to calculate throughput. LTE System Toolbox™ provides a convenient framework in which you can experiment with different modulation and demodulation algorithms and compare their characteristics. Figure 11 shows how this can be achieved in a few lines of MATLAB code.

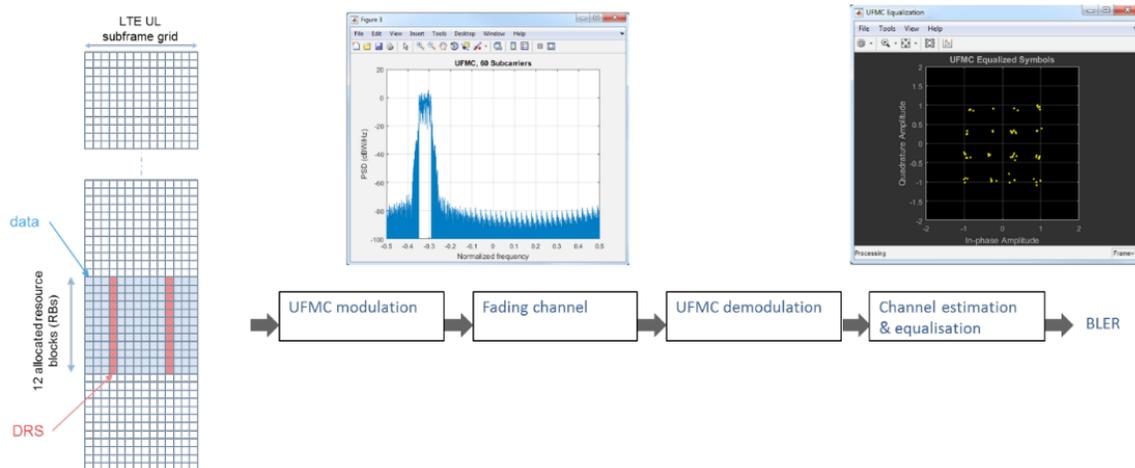


Figure 10. Framework for simulating and measuring impact of UFMC vs. LTE baseline performance.

```

for symbolNr = 1:nrSymbolsPerSubf

    % Initialize signal for 1 PRB
    PRBsig = zeros(FFTLength+filterLen-1,1);

    for PRBnr = 1:numSubbands

        % Extract sub-band
        symbolsIn = txSubframe((PRBnr-1)*subbandSize+(1:subbandSize) ...
            +subbandOffset, symbolNr);

        % Apply IFFT
        offset = subbandOffset+(PRBnr-1)*subbandSize;
        symbolsInOFDM = [zeros(offset,1); symbolsIn ...
            ;zeros(FFTLength-offset-numel(symbolsIn),1)];
        ifftOut = ifft(iffshift(symbolsInOFDM));

        % filter sub-band
        filterOut = conv(RBFilter(:,PRBnr),ifftOut);
        PRBsig = PRBsig+filterOut;
    end

    txSig(ModSymbolLen*(symbolNr-1)+(1:ModSymbolLen)) = PRBsig;
end

```

Figure 11. MATLAB code for simulating UFMC modulation.

Case Study: Massive MIMO for Interference Reduction

Massive MIMO will be one of the critical technologies in 5G systems. One of its aims is to increase system capacity by reducing interference between users sharing the same resources. In the setup shown in Figure 12, three mobile devices share the same resources and are separated in space. This means that the downlink transmission from the base station is beamformed in the direction of the mobile devices. LTE System Toolbox is used to generate the LTE uplink waveforms, and Phased Array System Toolbox™ is used to implement the beamforming algorithms.

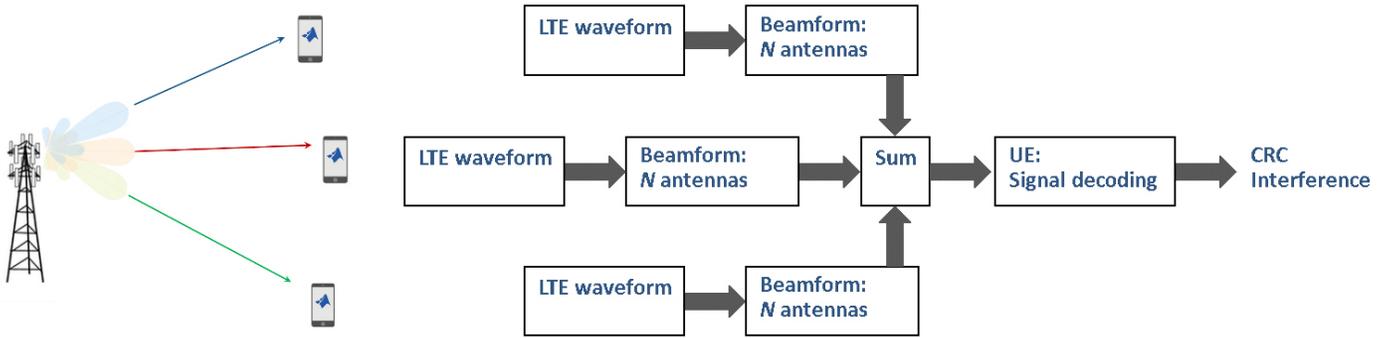


Figure 12. Multiple interfering mobile handset scenario and beamforming algorithm to mitigate interference.

In this example, the engineer can easily modify the number of antennas to be used in the array at the base station and study amount of energy leaked between users. The plots in Figure 13 show the radiation pattern for the three mobile devices and a measure of the interference between them. It is desirable to reduce the amplitude of the bars off the diagonal. Note how this is the case when more antennas are used. It is also clear how the beams in the radiation patterns are also narrower.

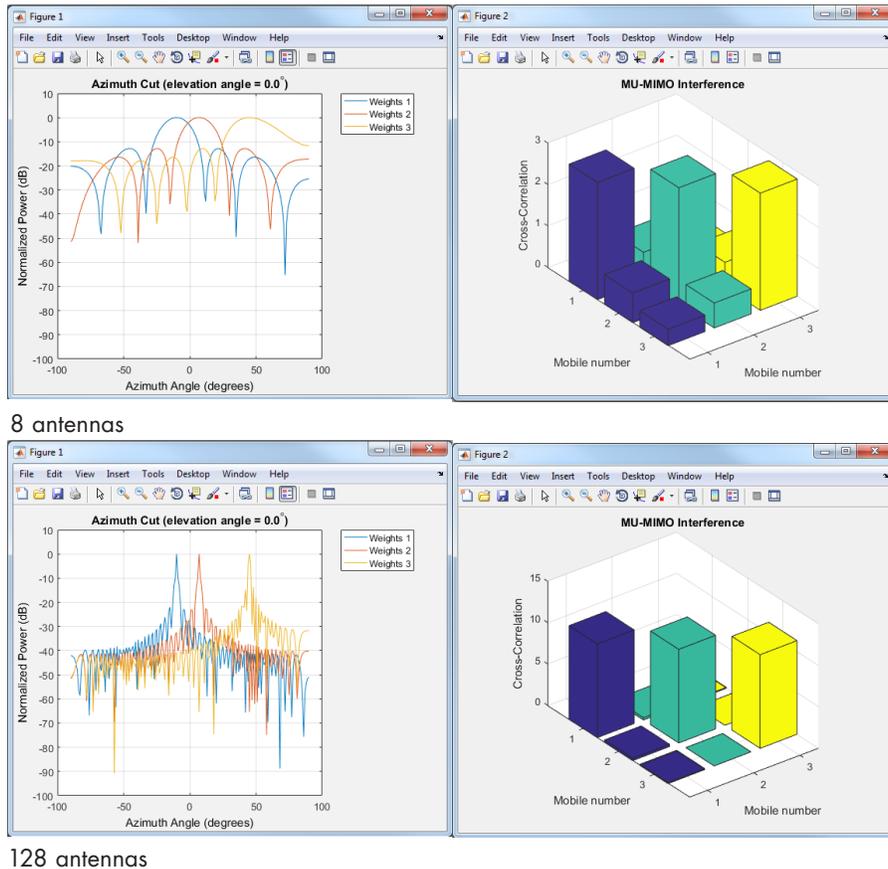


Figure 13. Beamforming patterns and multiuser interference for 8 and 128 antennas.

Joint RF-Baseband Modeling and Simulation

Traditionally, digital communications, RF, and antenna engineers have worked in different environments and integration occurred when the first hardware prototype was developed. The advent of highly integrated RF transceivers 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. 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
- Rapid prototyping using virtual test benches that make lab prototyping faster, more reliable, and easier to debug

This workflow reduces time spent in validating behavior and debugging test benches, and improves collaboration with colleagues, suppliers, and customers.

Case Study: Modeling a Tunable RF Transceiver

This example shows a model of an Analog Devices® AD9361 tunable RF receiver. The model represents the device architecture and includes the following:

- RF front end
- Analog-to-digital conversion (ADC)
- Digital down conversion (DDC)
- Control logic to automatically adapt the gain of the different stages

This model is architecturally accurate, and it is characterized by measurements, which makes the simulation results robust and accurate. The RF front end is modeled with the circuit envelope method, which guarantees fast simulation. The analog filters are programmable and modelled with behavioral continuous-time filters.

Nearly everything in the device is programmable, which makes it hard to test and program in the lab. For this reason, a model that accurately reproduces the chip behavior is very helpful to determine the optimal setup of the receiver.

Important elements of the AD9361 model (Figure 14) include:

- The multidomain model accurately describes a real device at the system-level and encompasses RF, analog and digital subsystems including programmable filters.
- The test bench enables testing the receiver with standard or with custom signals.
- The ADC's dynamic range and noise floor are accurately taken into account by means of a mixed-signal behavioral model.
- The AGC is modeled with a time-triggered state machine that takes into account the true system dynamics.
- The model reproduces the measurements performed on the actual device.

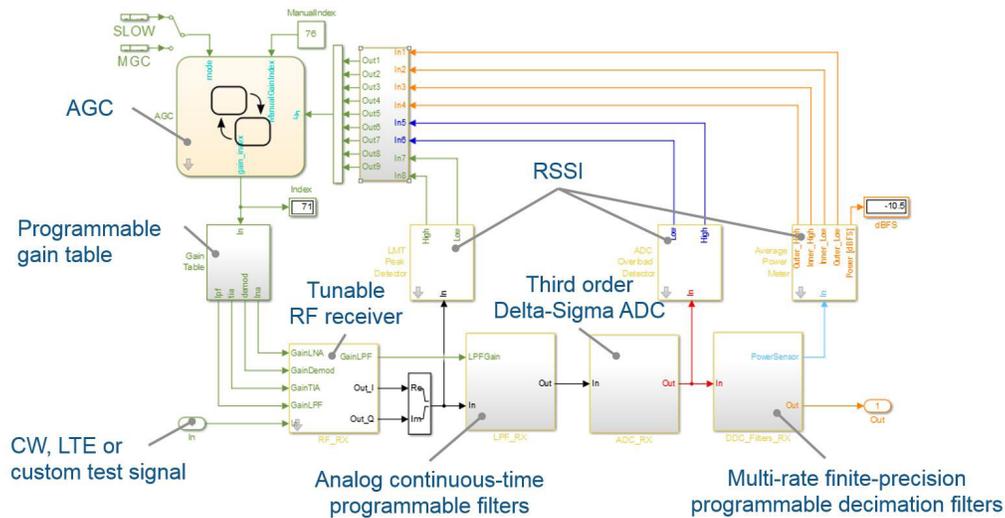


Figure 14. Architecturally accurate model of the Analog Devices AD9361, with RF, analog, DSP, and state machine components.

With this model, you can use CW tones to test the receiver, or wideband standard-compliant LTE signals (5, 10, 20MHz). These are I/Q baseband signals and they describe the envelope, or the modulated spectrum of the test waveforms. Thanks to the modeling strategy, the model is fast to simulate, and allows streaming LTE frames in just minutes of simulation because it simulates approximately 1 ms of data in 20 seconds.

Figure 15 illustrates a simulation that measures the impact of an RF receiver model on the end-to-end performance of an LTE design. In this example, an LTE test signal created in MATLAB stimulates both a MATLAB LTE reference model with an ideal RF front end and a comparable model in Simulink that includes an architecturally accurate RF receiver. The output of the two models is compared using EVM measurements and spectrum visualization.

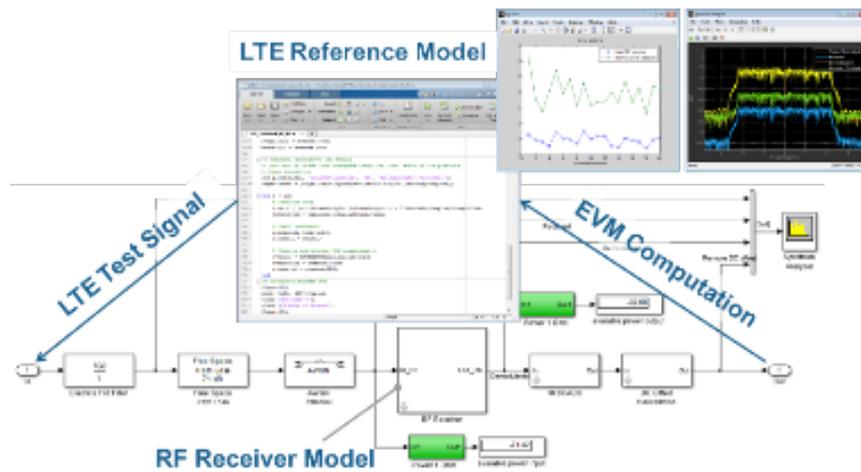


Figure 15. Testing the RF receiver model with an LTE reference.

Design Verification and Testing

Standard-Compliant Signal Generation and Measurement

Testing is a critical part of the wireless development process. In fact, it's 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 as well as scopes 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 with LTE System Toolbox or WLAN System Toolbox™, you can:

- Generate a standard-compliant signal in just a few lines of MATLAB code or with a simple app
- Analyze test results with scopes and measurements such as EVM and ACLR
- Verify designs with simulations or live over-the-air tests

Figure 16 shows the code for generating a standard-compliant signal for an 802.11ac very high throughput (VHT) signal and frame structure, along with the resulting signal. Figure 17 shows a similar program for generating a standard-compliant LTE downlink signal. The same result can be achieved using the graphical app.

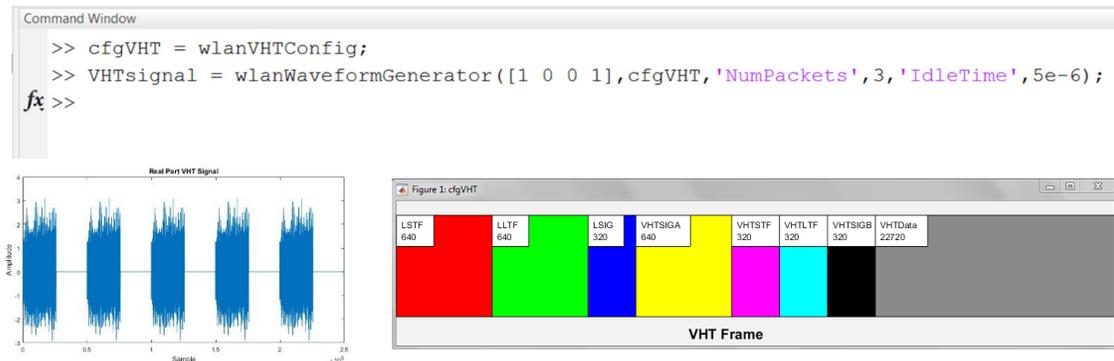


Figure 16. 802.11ac VHT signal generation.

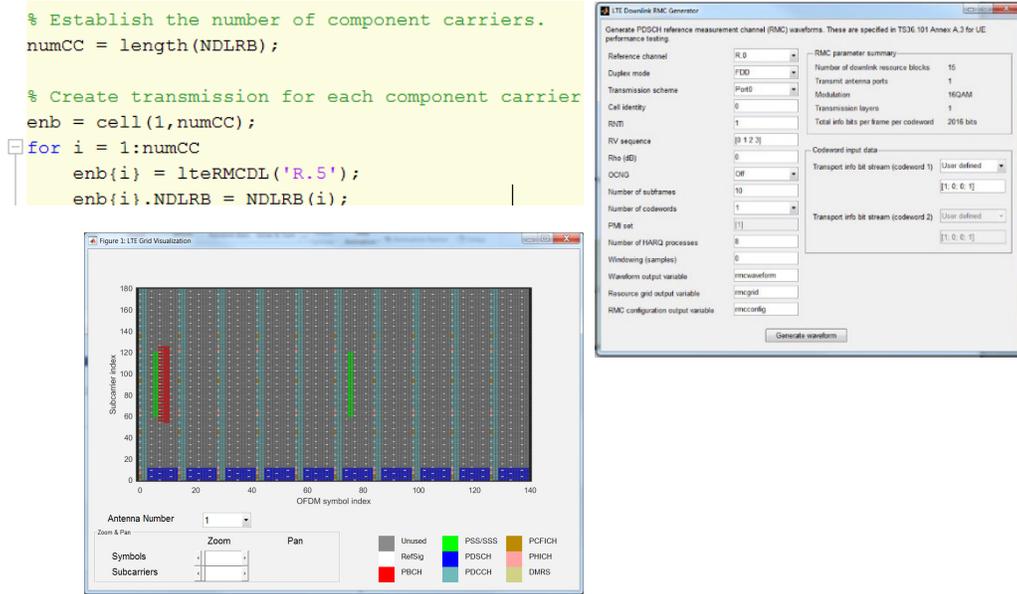


Figure 17. MATLAB code and app to perform LTE signal generation.

Signal analysis capabilities in MATLAB and communication products complement the signal generation to provide a complete testing framework. Figure 18 shows the use of these capabilities to evaluate the performance of an LTE receiver in a carrier aggregation scenario. The simulation aggregates three LTE signals and supports instrument-quality visualization and measurement of the output signals. Equivalent MATLAB functions perform the measurements programmatically, facilitating automated testing.

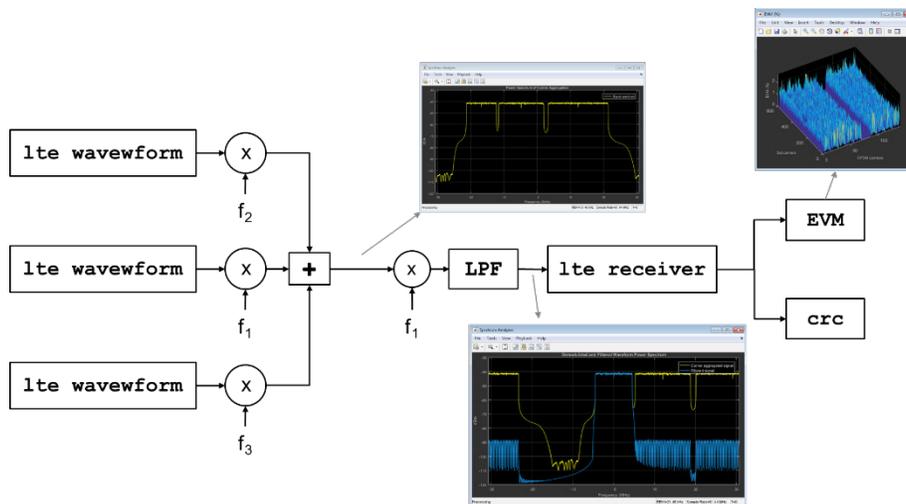


Figure 18. Measurement and visualization of an LTE receiver output in a carrier aggregation test.

Over-the-Air Testing

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, Anritsu, and other RF instrument vendors. SDR support packages are available for Zynq® and Avnet® PicoZed™ SDRs, Xilinx® FPGA Radio, USRP®, and RTL-SDR. From these options, you can choose the hardware that's most appropriate for your requirements and budget (Figure 19).

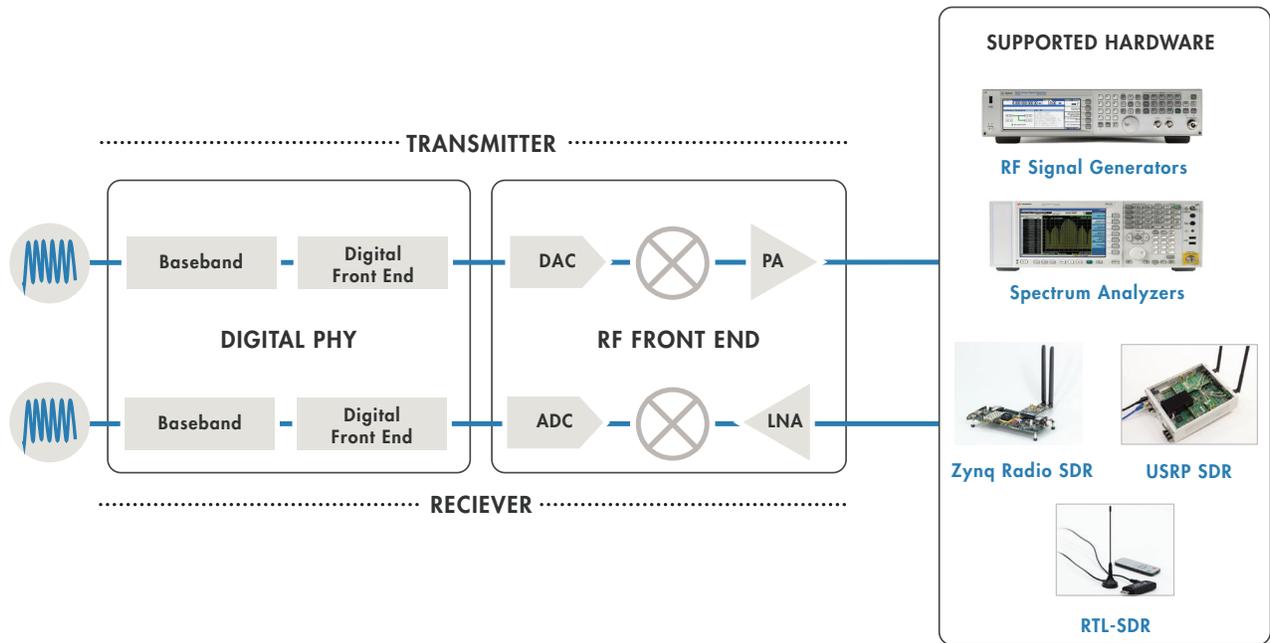


Figure 19. MATLAB and Simulink support for over-the-air testing with a range of available SDR and RF instrument hardware.

Case Study: LTE Scanner

This example uses MATLAB and LTE System Toolbox with an SDR platform to scan a frequency band for an available LTE signal from nearby base stations, as depicted in Figure 20, and decode cell-specific information.

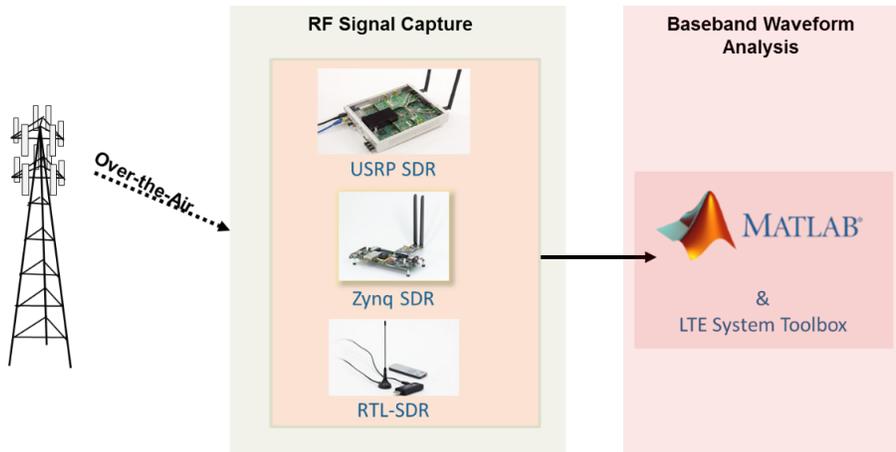


Figure 20. Using SDR hardware with MATLAB to scan for an LTE signal. The analysis is hardware-independent and can be used with any of the depicted radio platforms.

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 System Toolbox to perform the cell search procedure. The scanner can decode the master information block (MIB) and the system information block 1 (SIB 1) to provide information about the found cells.

The app shown in Figure 21 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.

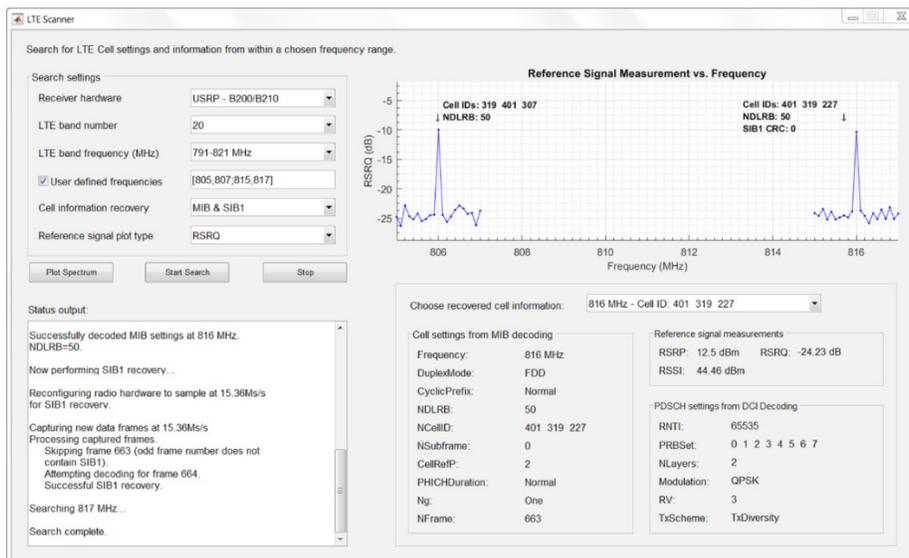


Figure 21. LTE cell scanner app.

Model-Based Design for Wireless Prototyping and Implementation

Prototyping Workflow for FPGA and SoC-Based SDR Platforms

You can employ Model-Based Design to migrate a wireless transmitter or receiver design from a simulation model to the target hardware. Once you have a floating-point design running in MATLAB or Simulink that has been validated with over-the-air tests, you can iteratively convert algorithms to hardware-ready fixed-point Simulink models and deploy the design to the target system.

As you prototype each element of the design, the rest of it remains in the loop on your PC to serve 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 22 shows the incremental process from pure simulation to RF testing.

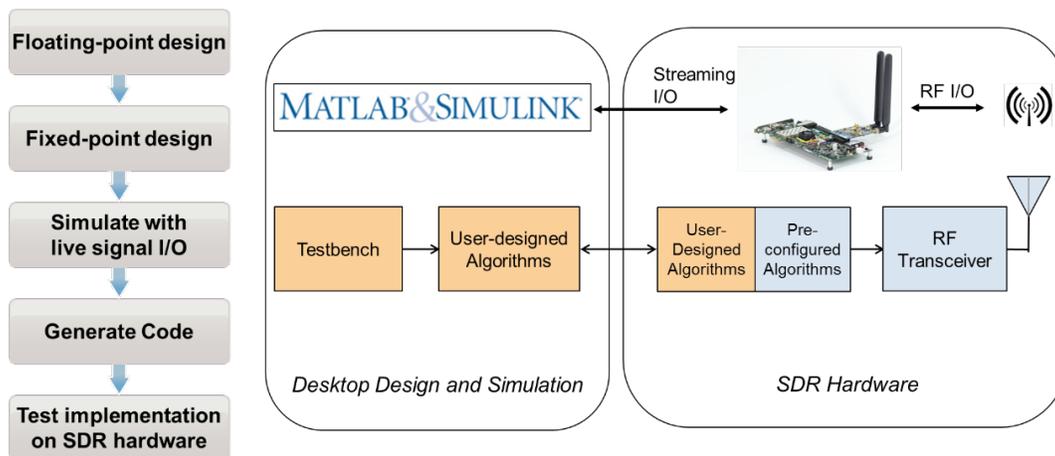


Figure 22. Workflow for migration of algorithms from simulation to hardware. User-design components are shown in orange, and the preconfigured SDR framework is shown in blue.

The steps in the workflow include:

- Partitioning the Simulink model into functions that will target the FPGA fabric and the ARM® processing system on the Zynq SoC
- Refining the Simulink model to achieve desired performance of the generated HDL code
- Generating the source HDL and C code for the receiver algorithm
- Integrating and testing the generated source code in the radio platform design.

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.

SDR hardware such as the Avnet PicoZed SDR with a Zynq SoC and an Analog Devices AD9361 enables you to completely replicate a simulation of the signal chain from RF to base-band in the SDR hardware.

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 for multirate filtering, PAPR suppression, digital predistortion, and baseband processing.

Figure 23 shows a Simulink model of an LTE OFDM modulator and detector (top) and an HDL implementation-ready model of the detection algorithm. 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.

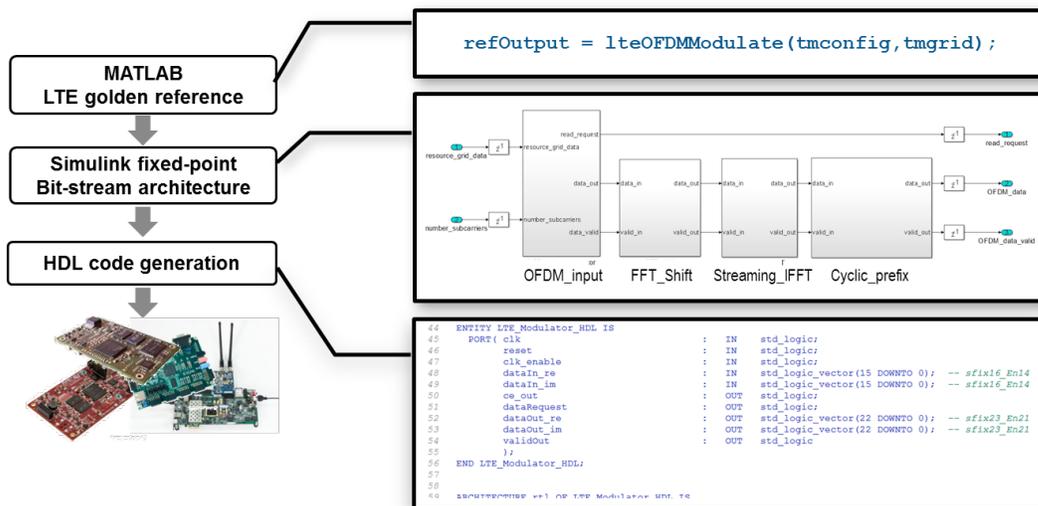
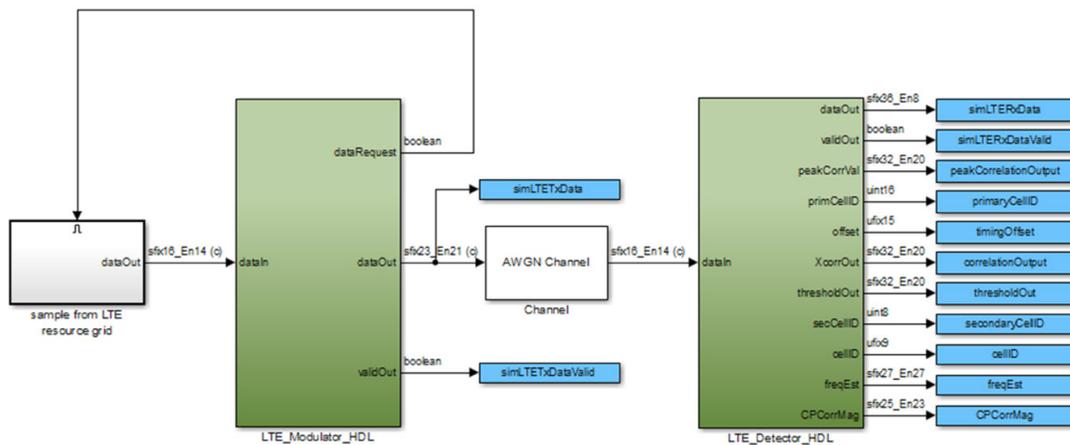


Figure 23. Simulink model for and LTE-compliant modulator and detector (top), and workflow for designing and verifying an HDL implementation of the modulator subsystem (bottom).

Summary

The race to innovate and keep pace with advances in wireless technology is also 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 re-spins, 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.

Learn More About Wireless Design with MATLAB

- Software-Defined Radio
- RF Systems
- 5G Wireless Technology Development
- Beamforming

Read How Leading Companies Are Using MATLAB for Wireless Design

- DigitalGlobe Simulates Complete Satellite-to-Ground Communications Systems
- DOCOMO Beijing Labs Accelerates the Development of Mobile Communications Technology
- Harman Becker Designs and Verifies OFDM Radio Receivers
- IDT-Newave Reduces Semiconductor Design Time by Months
- AT4 wireless Increases Internal Test Coverage to Over 90% for LTE Physical Layer Test Equipment Designs