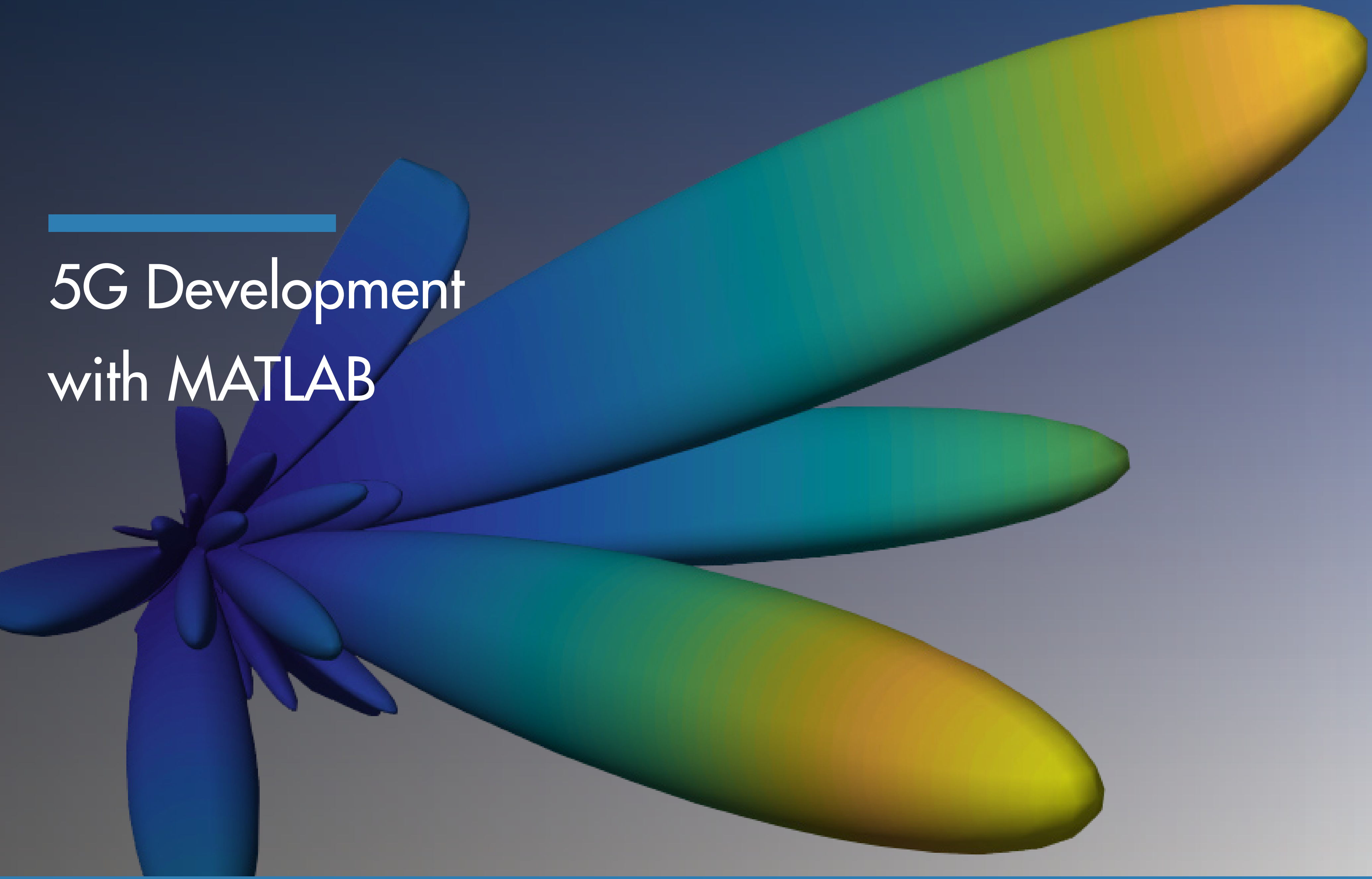
5G Development with MATLAB

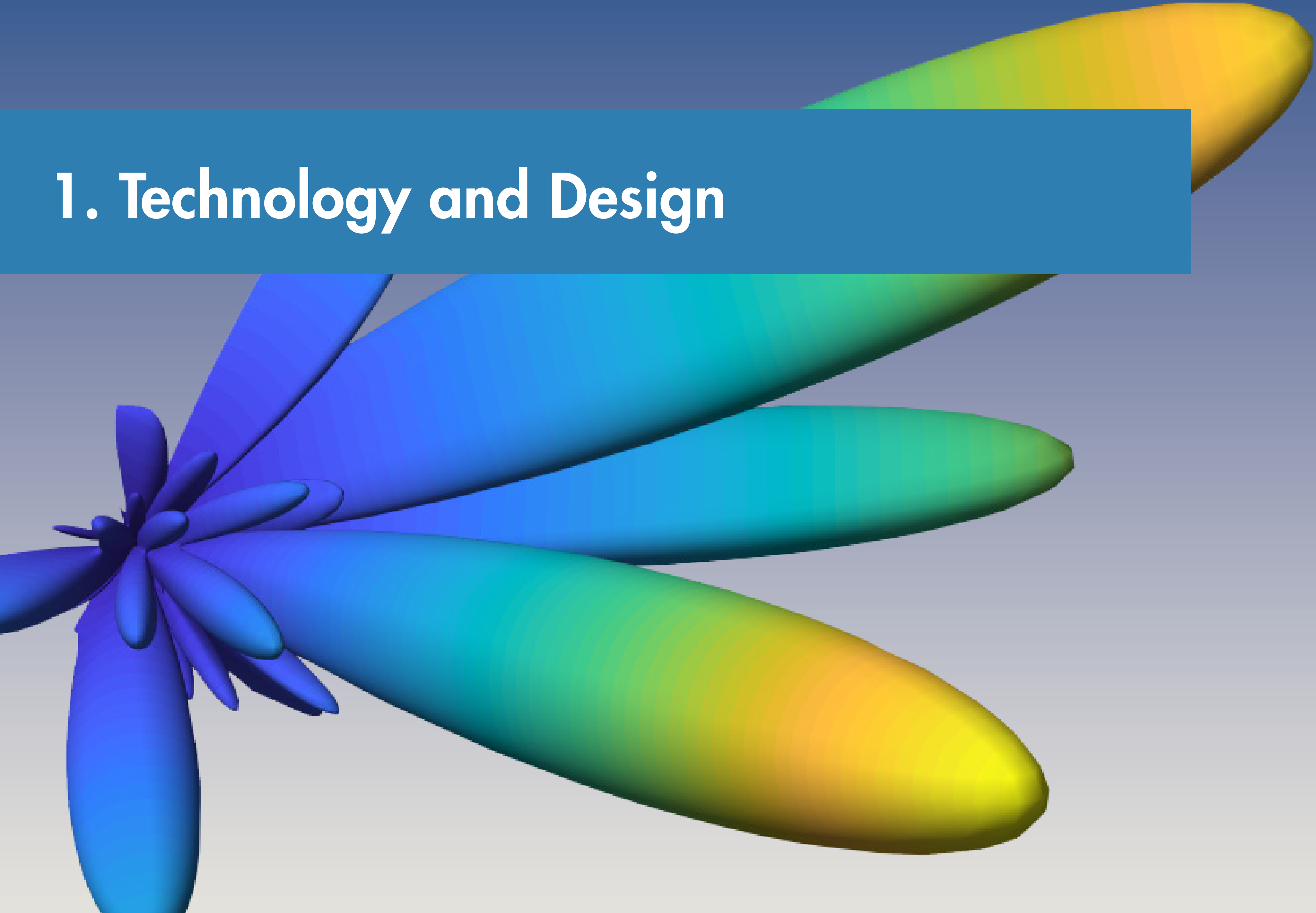


MathWorks®

5G Development with MATLAB

1. Technology and Design
2. New Architectures and Algorithms
3. Accelerating Prototypes and Field Trials
4. System Verification and Testing

1. Technology and Design



What is 5G?

5G (5th generation wireless systems) is the next major phase of mobile telecommunications standards. The scope of 5G will ultimately range from mobile broadband services to next-generation automobiles and connected devices.

The initial 5G New Radio (NR) specification was completed in June 2018 and published in the 3GPP Release 15 specification. Now, a variety of industry players, including network equipment vendors, network operators, semiconductor vendors, and device manufacturers, are developing new products that implement the new standard.

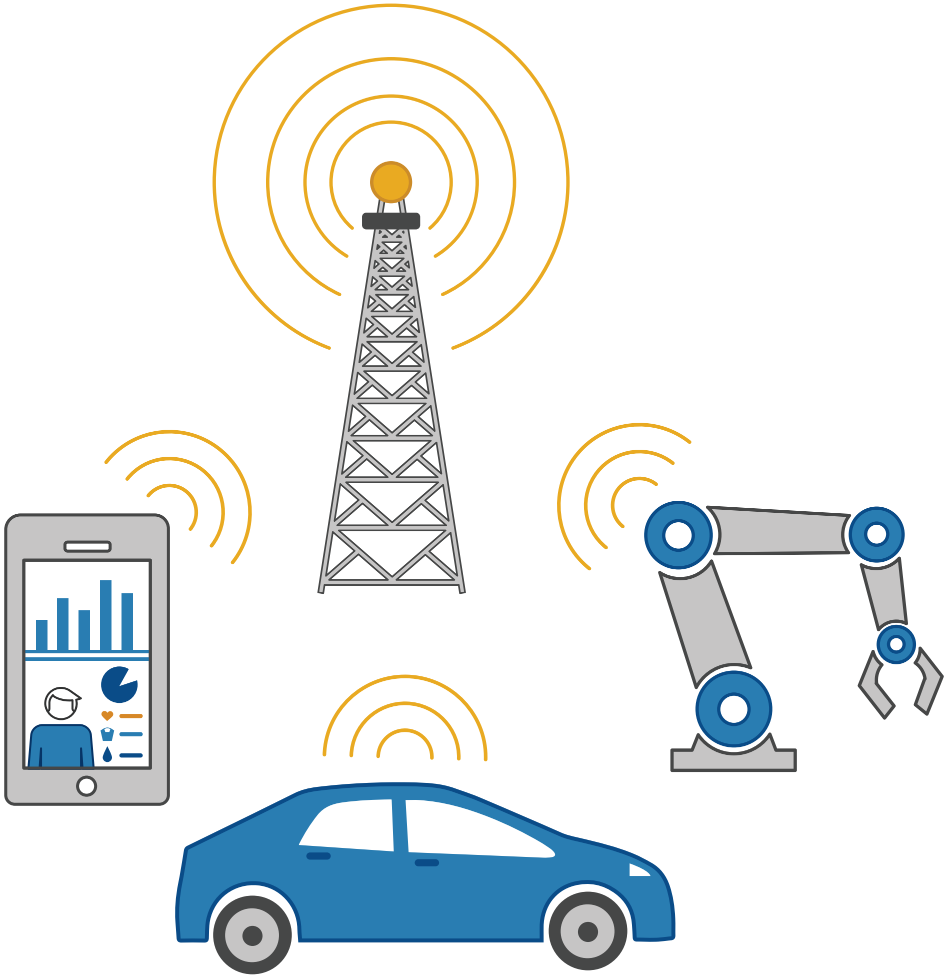
If you are already familiar with 5G, feel free to skip ahead to sections 2–4, which discuss strategies for doing 5G design and development with *MATLAB®*: new algorithm design (Section 2), accelerating prototyping and field trials (Section 3), and system verification (Section 4).

What's driving 5G?

Two major trends are behind the race to 5G: the explosive growth in demand for wireless broadband that can carry video and other content-rich services, and the Internet of Things (IoT), where large numbers of smart devices communicate over the Internet. To achieve these objectives, 5G will provide **extreme broadband speed, ultralow latency**, and **ultrareliable web connectivity**.

5G networks and devices will require substantially different architectures, radio access technology, and physical layer algorithms. Dense networks of small cells will complement macro base stations, operating at millimeter wave technologies and employing massive MIMO antenna arrays. And the processing components within network equipment and user devices will become more integrated and adaptive.

Innovations like hybrid beamforming are stretching the old ways of developing wireless systems. These highly integrated technologies require a corresponding integration of engineering domain expertise and tools.



5G Terminology and Expected Applications

By providing higher bandwidth capacity than current 4G–supporting broadband, 5G will enable a higher density of mobile broadband users and support ultrareliable device-to-device and massive machine-type communications.

Key 5G Parameters	
Latency in the air link	<1 ms
Latency end-to-end (device to core)	<10 ms
Connection density	100x vs. current 4G LTE
Area capacity density	1 (Tbit/s)/km^2
System spectral efficiency	10 (bit/s)/Hz/cell
Peak throughput (downlink) per connection	10 Gbit/s
Energy efficiency	>90% improvement over LTE

eMBB—Enhanced Mobile Broadband

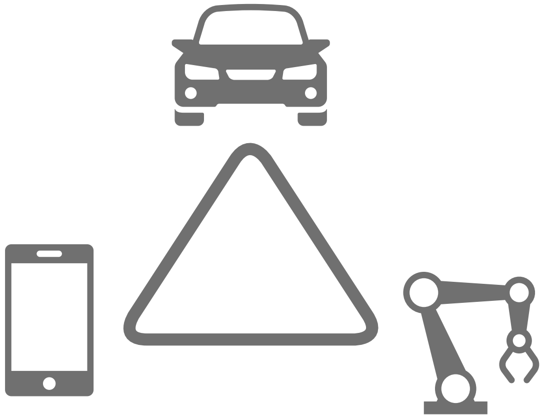
- For high-capacity and ultrafast mobile communications for phones and infrastructure, virtual and augmented reality, 3D and ultra-HD video, and haptic feedback

URLLC—Ultrareliable and Low Latency

- For vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications, autonomous driving

mMTC—Massive Machine-Type Communications

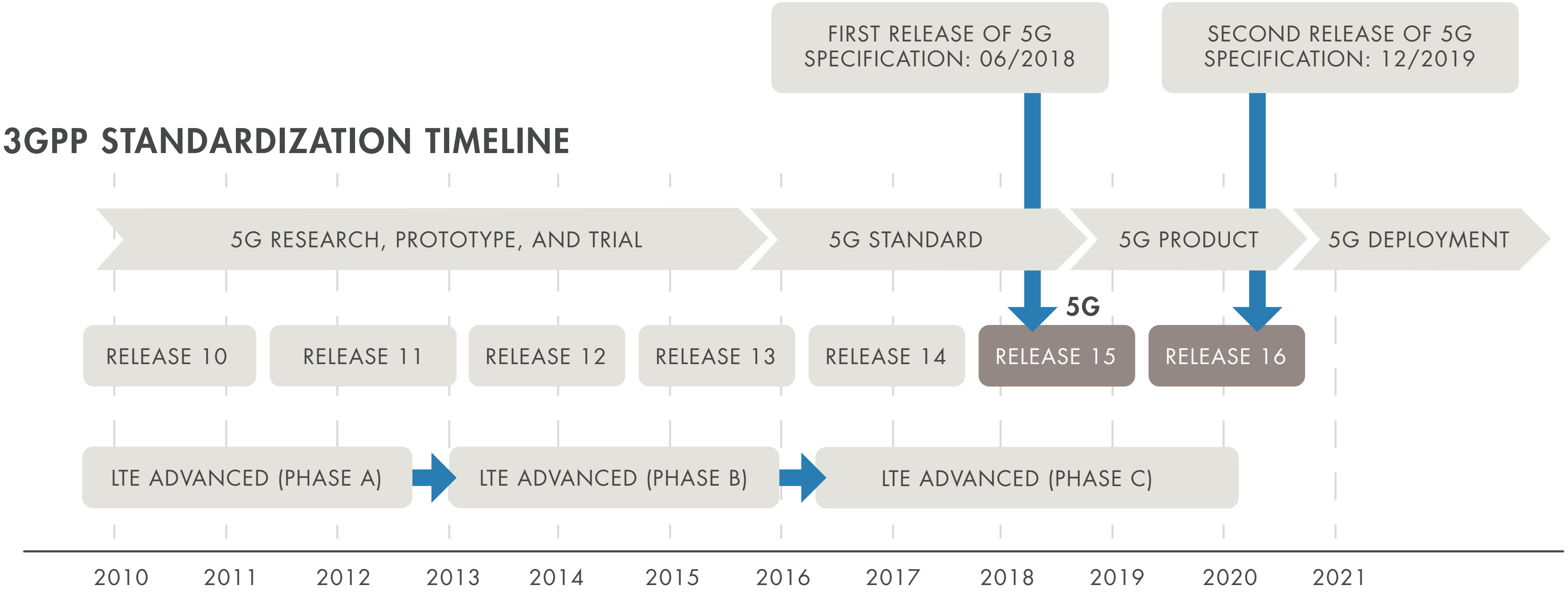
- For consumer and industrial IoT, Industry 4.0 mission-critical machine-to-machine (MC-M2M)



5G Standard: Timeline

The 3GPP standardization group defines the wireless 5G standard, with help from many participants and contributors around the globe. Release 15 of the 3GPP standard, finalized in June 2018, introduces the 5G standard. Physical layer algorithms, millimeter wave technology, and massive MIMO architectures are expected to be significantly different from 4G LTE technologies.

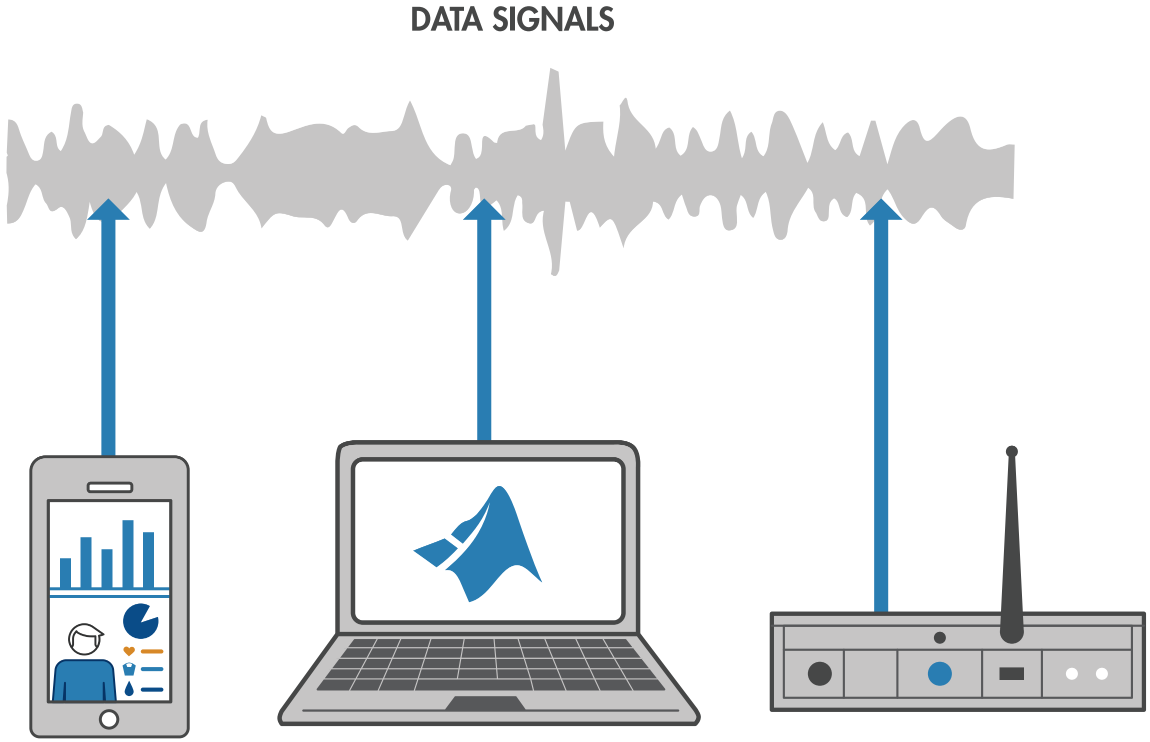
Current 5G development focuses on enabling technologies such as flexible baseband and RF technologies, hybrid beamforming, and massive MIMO systems; rapid prototyping and field trials; and verification of compliance with the new standard specifications.



A New Physical Layer for 5G

The 5G physical layer will depart from 4G LTE in a number of ways, in order to improve spectral efficiency and data rates. One distinctive feature is a significant jump in the number of active antennas and antenna arrays, and the related issues of beamforming and millimeter wave RF signal processing. New modulation and coding schemes, power and low-noise amplifier designs, and channel models all need to be developed.

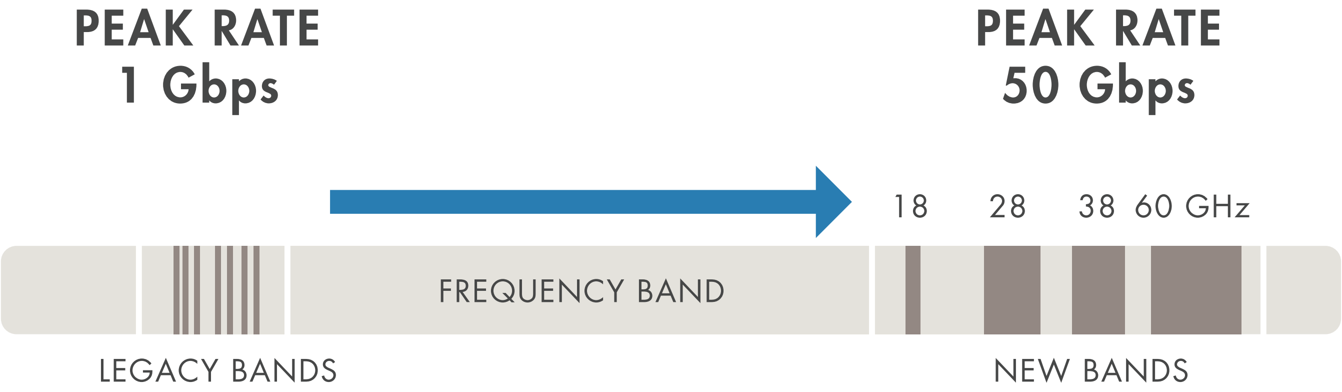
Sections 2–4 of this ebook discuss 5G design and development strategies in detail: new algorithm design (Section 2), fast prototyping (Section 3), and system verification and field trials (Section 4).



Millimeter Wave Higher-Frequency Operation

Higher data rates (multi-Gbps) drive the need for greater bandwidth systems, and the available bandwidth in the spectrum up through 6 GHz is not sufficient to satisfy these requirements. (For reference, current cellular operation is below 3 GHz.) This has moved the target operating frequency bands up into the millimeter wave (mmWave) range for the next generation of wireless communication systems. For example, equipment developers have performed 5G NR trials with mobile network operators to demonstrate the viability of higher frequency transmissions. Those trials operate in the midband spectrum from 3.3 GHz to 5.0 GHz, as well as the mmWave spectrum at 28 GHz and 39 GHz, showcasing the unified 3GPP-based 5G NR design across diverse spectrum bands.

High frequencies will provide larger bandwidth availability and smaller antenna dimensions for a fixed gain, or higher gain for a given antenna size. However, this increases modem complexity in baseband and RF designs. To study the performance, we also need an accurate channel model for the new frequencies in 5G.



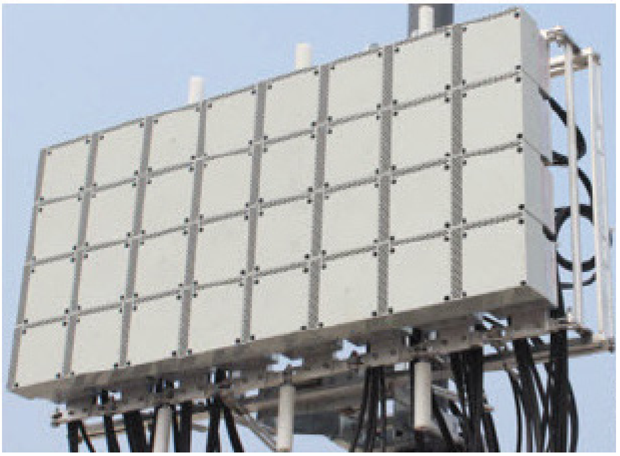
Massive MIMO: More Antennas

Another key technology for achieving greater spectral efficiency is **massive MIMO**. Massive MIMO, sometimes referred to as large-scale MIMO, is a form of multiuser MIMO in which the number of antennas at the base station is much larger than the number of devices per signaling resource. The large number of base station antennas relative to user devices results in a channel response that is quasi-orthogonal and has the potential to yield huge gains in spectral efficiency.

Designers face a challenge when scaling the number of antennas to hundreds:

- The **simulation speed** of traditional antenna design tools is slow for large antenna arrays.
- It is **difficult to simulate** the antenna coupling.
- **Hybrid beamforming** is needed to optimize the number of the RF chains.

See Section 2 for details on design architectures and algorithms.

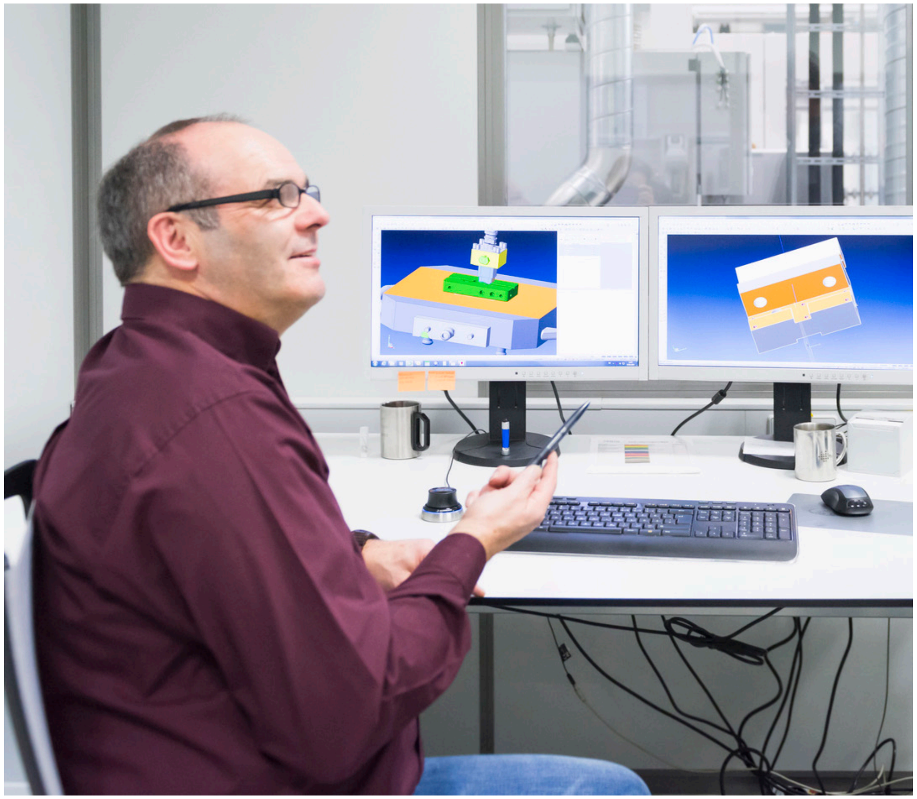


A massive MIMO antenna array for a Huawei 5G field trial.

Fast Prototyping of 5G Systems with Hardware Testbeds

Engineers working on 5G designs have realized the value of rapid design iterations and of placing proof-of-concept prototypes in field trials quickly. Hardware testbeds employed as flexible and reconfigurable design platforms have proven to be dependable and efficient for the rapid design and verification of new concepts, as well as for their deployment in precommercial field trials. The tools and workflows that interface to testbeds must support rapid design iterations and rapid deployment of new algorithms or design changes.

A fast prototyping workflow is discussed in Section 3.



Learn More

Ready for a deeper dive? Explore these resources to learn more about 5G technology developement and design.

Watch

Design 5G Wireless Technologies with MATLAB and Simulink Chalk Talk (32:00)

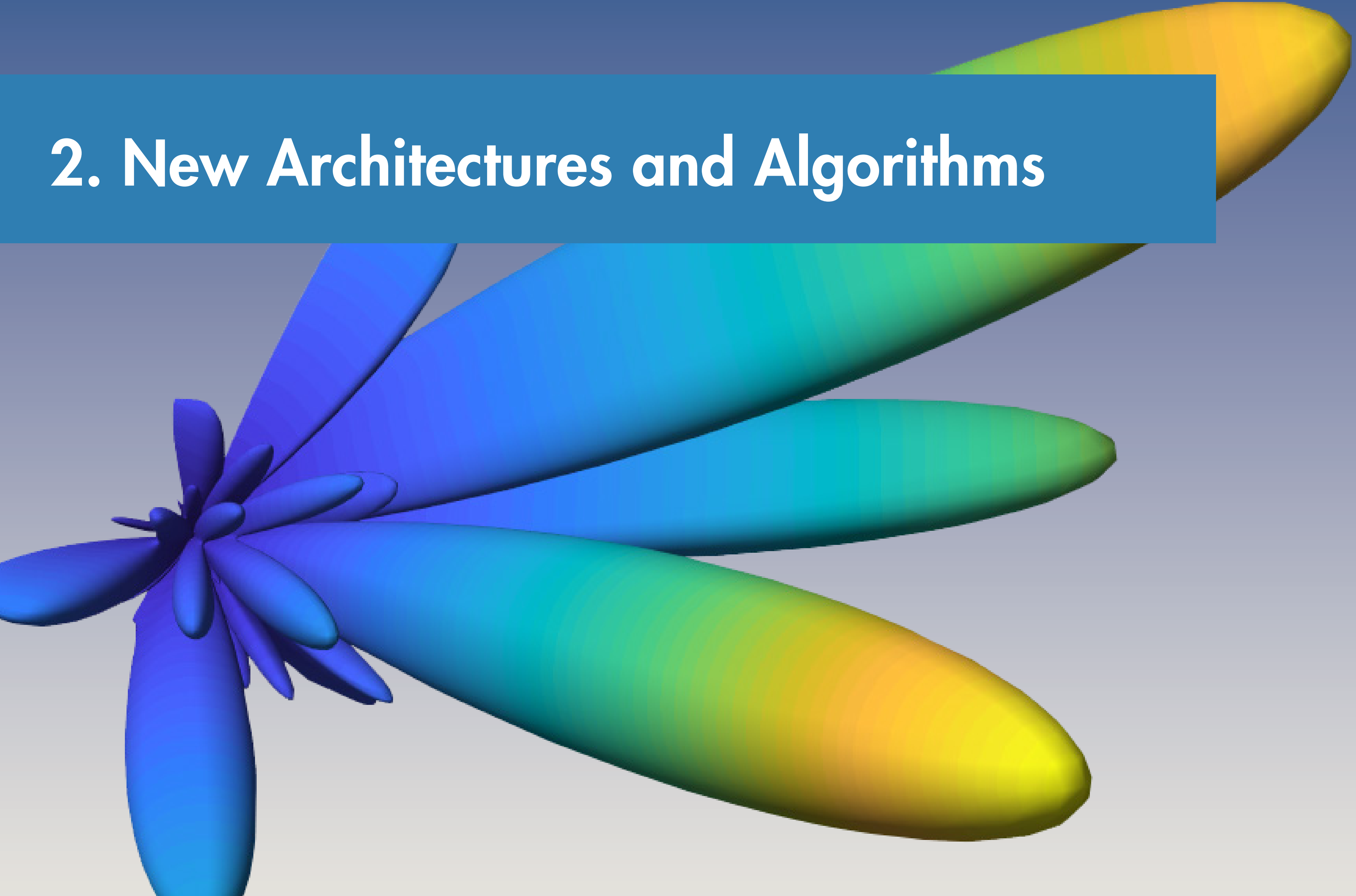
Explore

5G Wireless Technology Development

Read

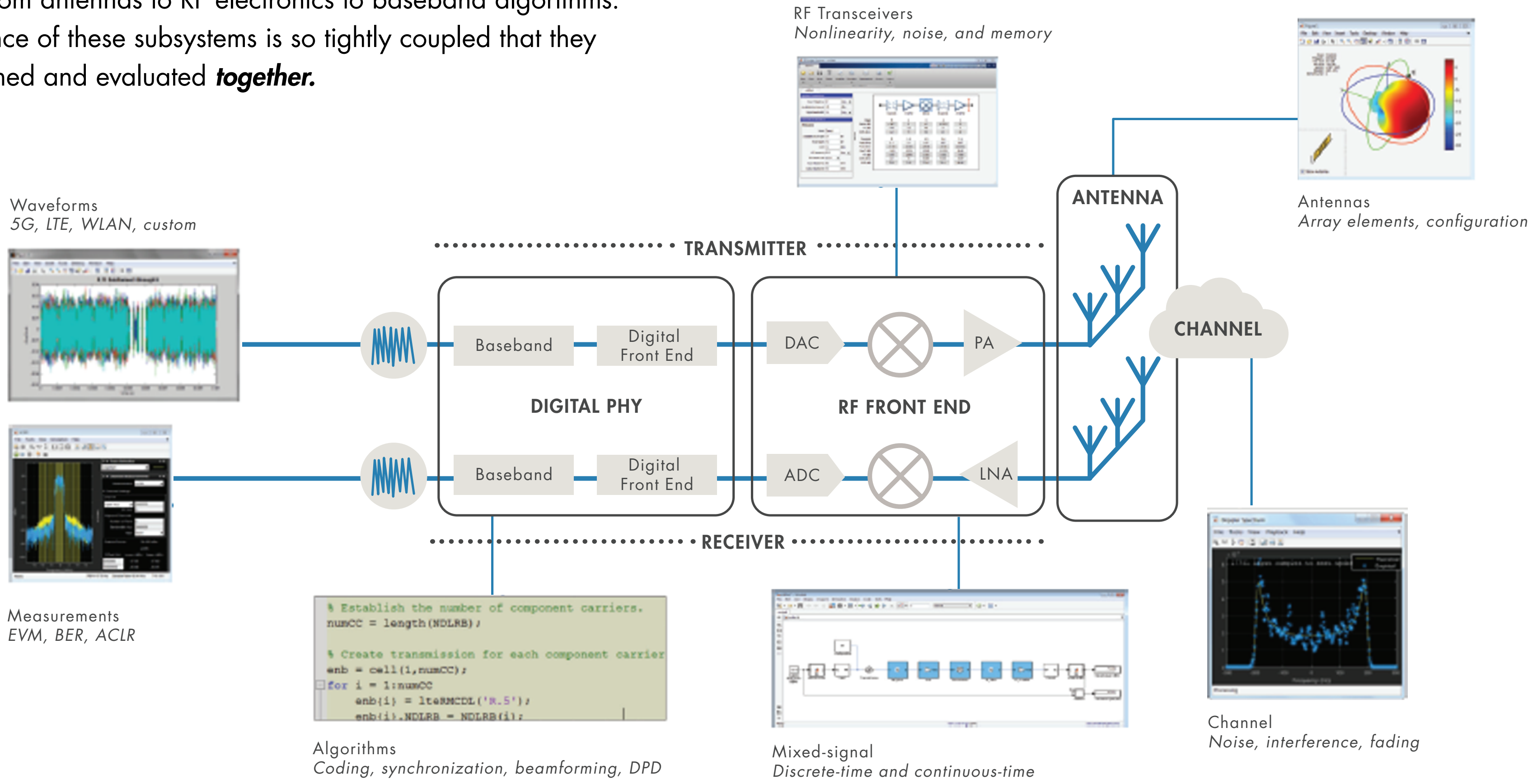
Advancing the 5G Wireless Standard at Convida Wireless: An Insider Look

2. New Architectures and Algorithms



New Design Architectures and Algorithms for 5G

The leap in 5G broadband speeds will be enabled by massive MIMO communication in the millimeter wave (mmWave) frequency range and by new radio algorithms that achieve more efficient use of spectrum. New design architectures and algorithms will affect every aspect of 5G systems, from antennas to RF electronics to baseband algorithms. The performance of these subsystems is so tightly coupled that they must be designed and evaluated ***together.***



RF Transceivers
Nonlinearity, noise, and memory

Antennas
Array elements, configuration

Waveforms
5G, LTE, WLAN, custom

Measurements
EVM, BER, ACLR

```
% Establish the number of component carriers.
numCC = length(NDLRB);

% Create transmission for each component carrier
enb = cell(1,numCC);
for i = 1:numCC
    enb{i} = lteRMCDL('R.5');
    enb{i}.NDLRB = NDLRB(i);
```

Algorithms
Coding, synchronization, beamforming, DPD

Mixed-signal
Discrete-time and continuous-time

Channel
Noise, interference, fading

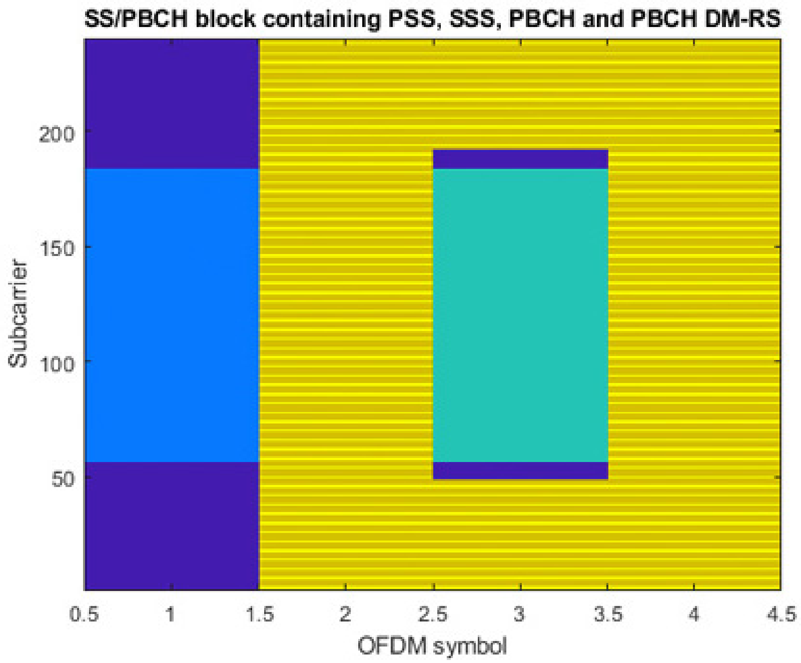
New 5G Physical Layer Algorithms for Greater Flexibility and Spectral Efficiency

The 5G wireless communication standard will provide significantly higher mobile broadband throughput with its enhanced mobile broadband (eMBB) mode. Several techniques and features have been standardized for 3GPP Release 15.

Among the key elements of 5G NR in 3GPP Release 15 are:

- Shorter slot durations, corresponding to increased subcarrier spacing, for increased signal bandwidth and shorter latency
- New channel coding methods such as LDPC for data and polar codes for control information, for more efficient error correction and improved data rates
- Spatial channel models for operation at current (<6 GHz) and mmWave (>28 GHz) frequencies

These elements have the potential to improve bandwidth, latency, and system efficiency, but they can add complexity and delay to your design. New tools available for use in MATLAB and Simulink let you quickly characterize the new features of 5G NR and make critical design tradeoff questions at an early stage.



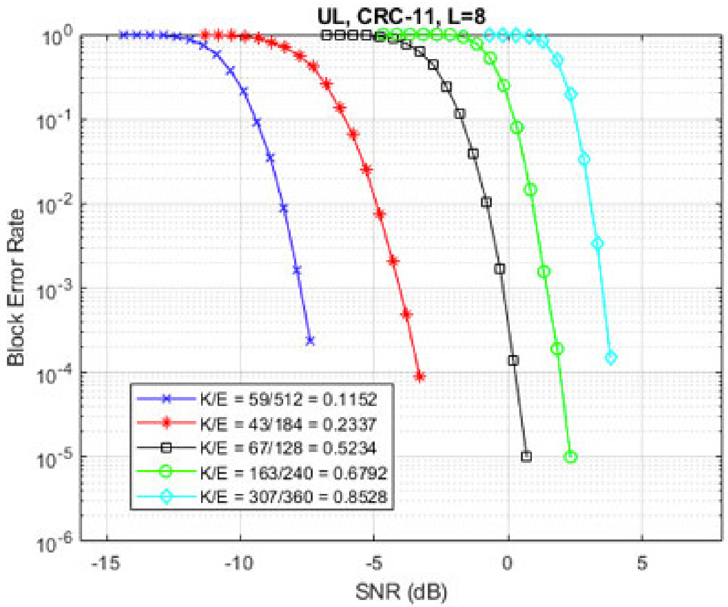
Detailed view of 5G SSB resource grid including primary and secondary synchronization sequences, broadcast channel, and associated DMRS.

New 5G Physical Layer Algorithms for Greater Flexibility and Spectral Efficiency *continued*

To help engineers developing 5G standard-compliant products and researchers exploring future 5G technologies, MathWorks released *5G Toolbox™* in September 2018. 5G Toolbox provides:

- **3GPP 5G standard-compliant OFDM waveform generation** using variable NR subcarrier spacing and frame numerologies
- **Downlink channel and signals,** including transport and physical channels (including CORESET support and transport, physical, and control channels)
- **New coding schemes** such as LDPC for data and polar codes for control information, for error correction and improved data rates
- **Propagation channel models,** including tapped delay line (TDL) and clustered delay line (CDL) channel models as specified in 3GPP TR 38.901
- **Customizable link-level reference designs,** enabling you to generate 5G signals, simulate the end-to-end NR PDSCH link throughput, and perform synchronization procedures including MIB decoding in the context of PBCH beam sweeping

5G Toolbox functions are implemented as open, editable MATLAB code, so you can easily understand and customize algorithm implementations.

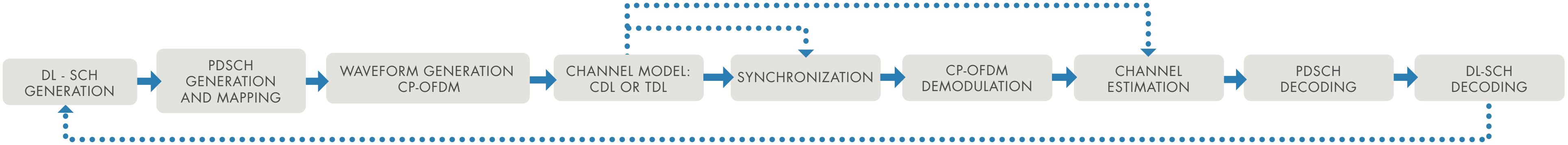


Results of block error rate simulations using polar coding.

New 5G Physical Layer Algorithms for Greater Flexibility and Spectral Efficiency *continued*

Using 5G Toolbox, you can measure the impact of different algorithms and design choices on system performance by simulating end-to-end system performance along with realistic 5G propagation channels.

5G WAVEFORM



DL - SCH GENERATION

PDSCH GENERATION AND MAPPING

WAVEFORM GENERATION CP-OFDM

CHANNEL MODEL: CDL OR TDL

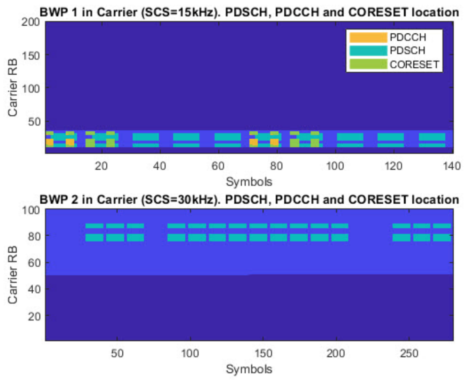
SYNCHRONIZATION

CP-OFDM DEMODULATION

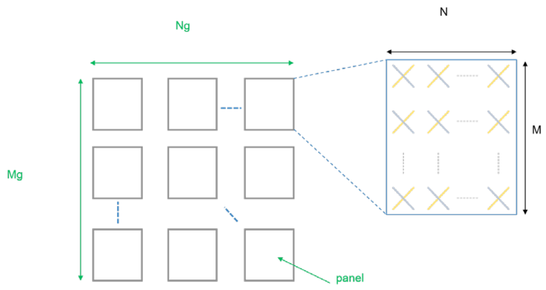
CHANNEL ESTIMATION

PDSCH DECODING

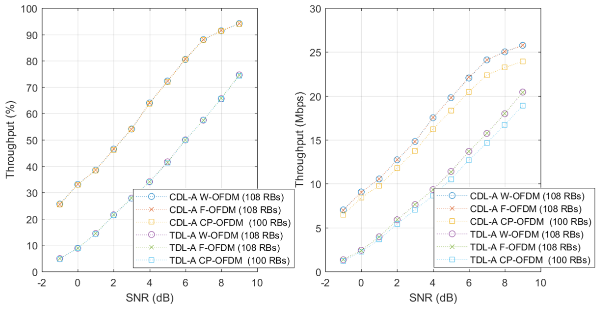
DL-SCH DECODING



5G Resource Grid



3GPP 38.901 Channel Model



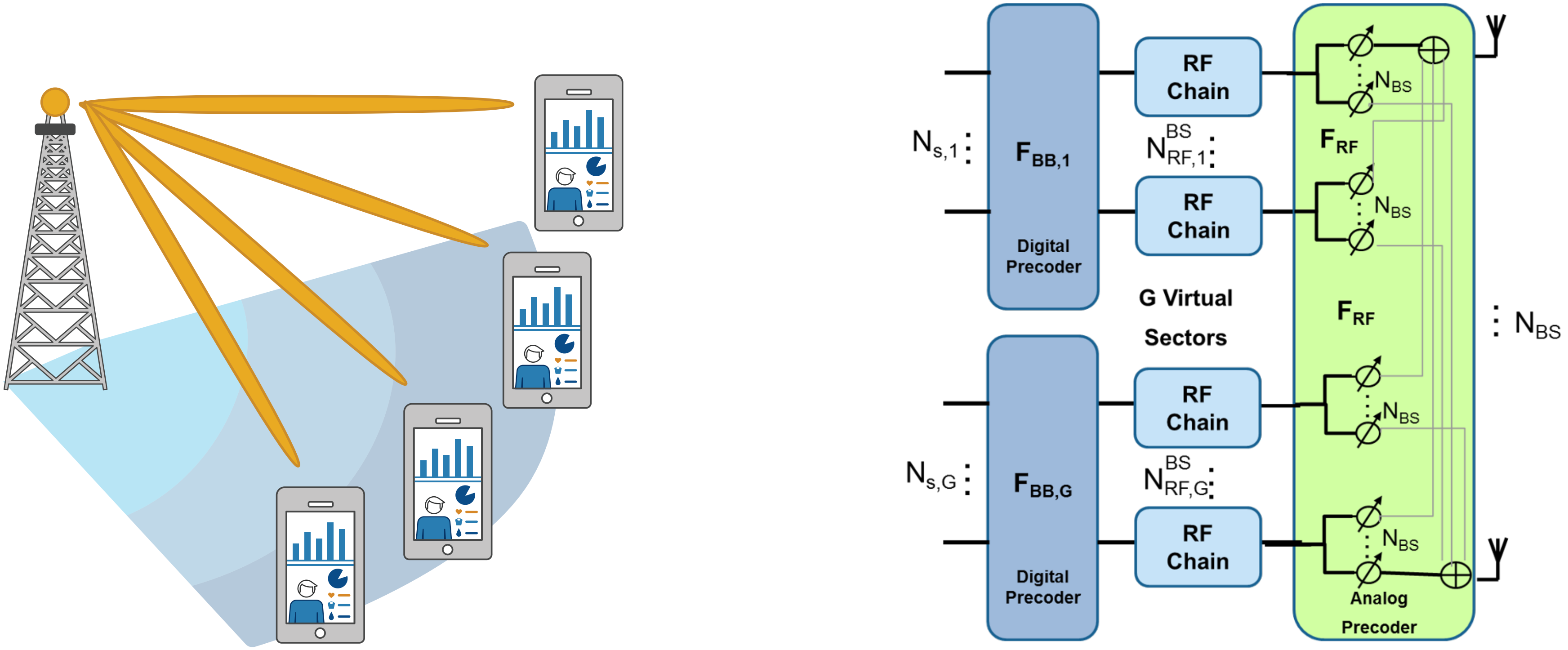
Throughput Measurement

Using a 5G downlink reference model, you can insert custom algorithms to simulate and optimize end-to-end link performance and validate standard compliance.

Massive MIMO Arrays for High-Gain Beamforming

5G mmWave designs require 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.

Having many antenna elements in a small area makes it practical to achieve a high beamforming gain. The highly directional beams help offset the increased path loss at the higher operating frequencies, because the beams steer power in a specific direction.



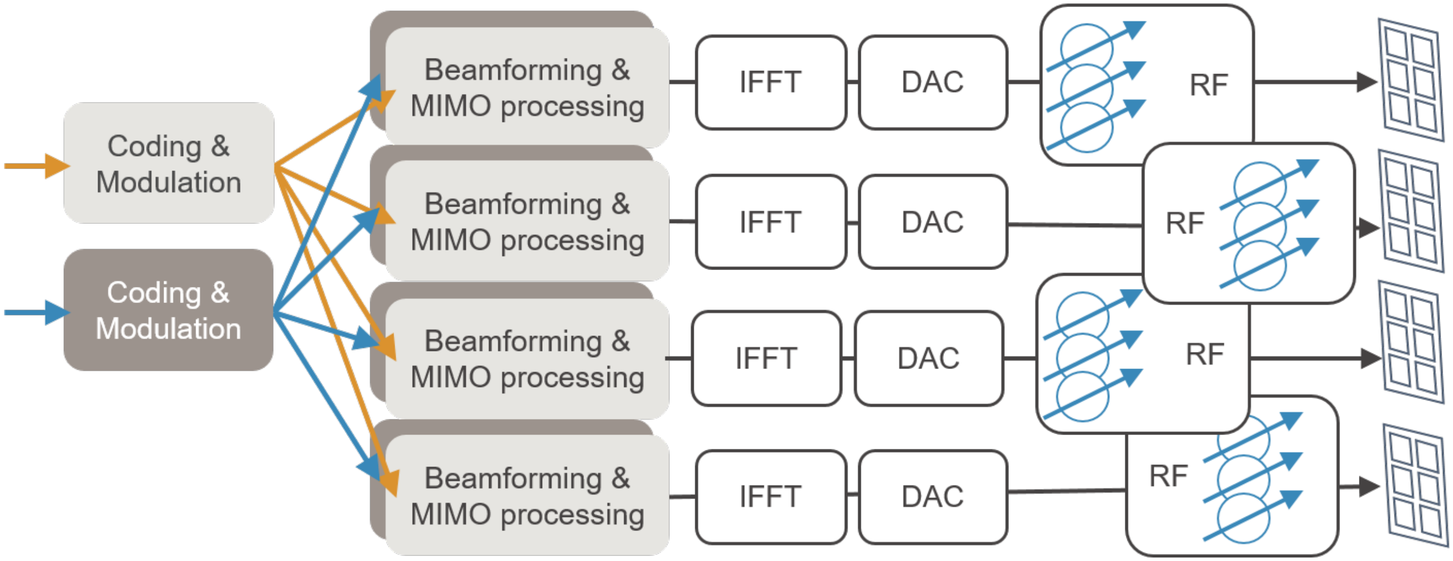
UEs in a group using beams with same elevation angle (left) and a hybrid beamforming array architecture (right).

Behavioral Simulation for Massive MIMO

Typical array designs include parameters such as array geometry, element spacing, the lattice structure of the elements, element tapering, and the effects of mutual coupling. By adjusting the design parameters, you can achieve tapering of the rows and columns of the array to reduce side lobe levels.

Achieving an optimal design thus 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 simulations.

Behavioral-level simulation of the antenna array system can address these challenges. Simulating at the behavioral level reduces the simulation time. This enables engineers to experiment with different array architectures and algorithms, simulate the performance of the array and associated algorithms, and iteratively adjust parameters to mitigate the effect of antenna coupling.



Coding & Modulation
Coding & Modulation
Beamforming & MIMO processing
Beamforming & MIMO processing
Beamforming & MIMO processing
Beamforming & MIMO processing
IFFT
DAC
RF
IFFT
DAC
RF
IFFT
DAC
RF
IFFT
DAC
RF

Massive MIMO antenna array design, which requires simulating the interactions between antenna, RF, and digital subsystems.

Optimizing Tradeoffs Between Antenna Gain and Channel Capacity

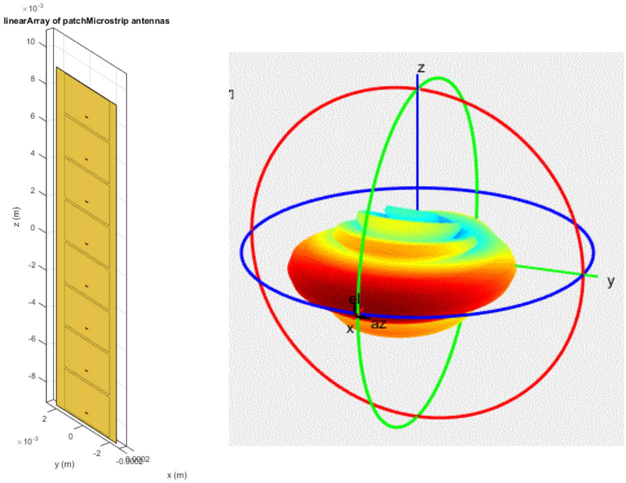
As a part of the design process, the larger antenna gains achieved with narrower beams must be balanced with the fact that MIMO systems are based on scattering environments that also depend on broader beam patterns to maximize channel capacity.

This is one of the key tradeoffs to assess, and it can be done iteratively with behavioral simulation.

The figures below show how you can develop and visualize behavioral simulations with *Phased Array System Toolbox™* and *Antenna Toolbox™* for MIMO array design tasks.

Configure the Initial Array

Use antenna element models, such as omnidirectionalor cosine elements, and rapidly move to more accurate analysis using patterns computed with electromagnetic (EM) tools or measured in the lab. Visualize the array performance characteristics, such as the radiation pattern shown below.

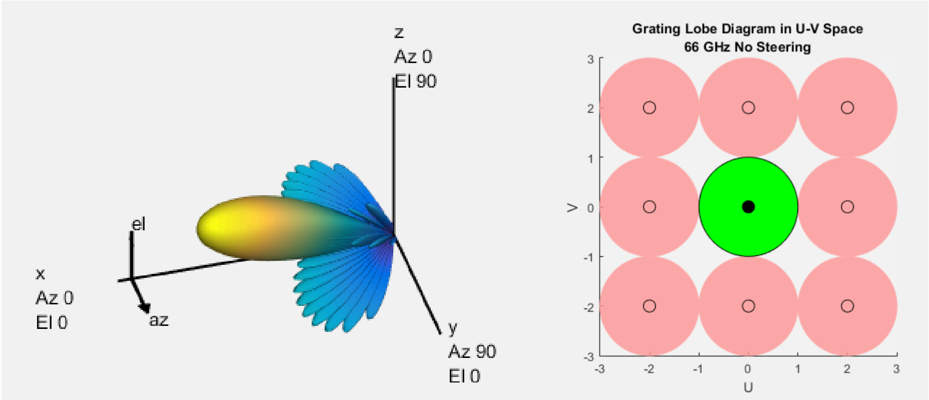


8x1 ULA subarray and corresponding radiation pattern.

Experiment Interactively with Design Parameters

Vary the array size, array geometry, element spacing, and tapering. Visualize the resulting geometry, 2D and 3D directivity, and the grating lobe diagrams.

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



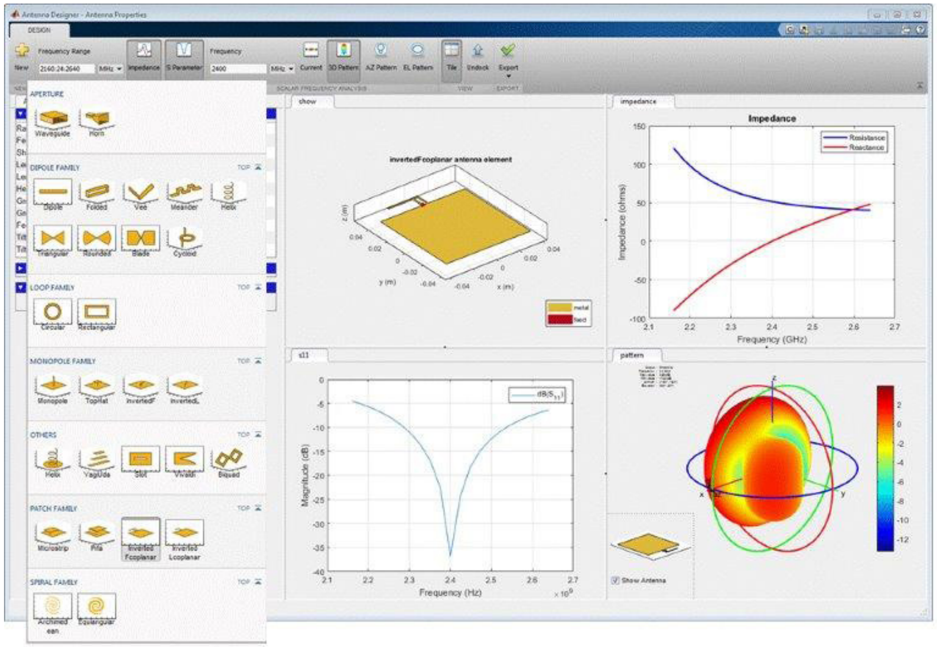
Beam pattern and grating lobe diagram for 66 GHz 64x64 element design.

Designing Antenna Array Elements

Detailed design of antenna elements, with appropriate free space patterns, can then be added to improve the fidelity of an antenna array model.

The figure at right illustrates an element pattern generated using a full wave EM solver in Antenna Toolbox. The toolbox uses the method of moments (MoM) algorithm to compute port properties such as impedance, surface properties such as current and charge distribution, and field properties such as the near-field and far-field radiation pattern.

You can use Antenna Toolbox to visualize antenna geometry and analysis results in 2D and 3D. You can also integrate antennas and array models into wireless systems, and use impedance analysis to design matching networks. The toolbox also provides radiation patterns for simulating beamforming algorithms.



Frequency Range 2160:24:2640 MHz

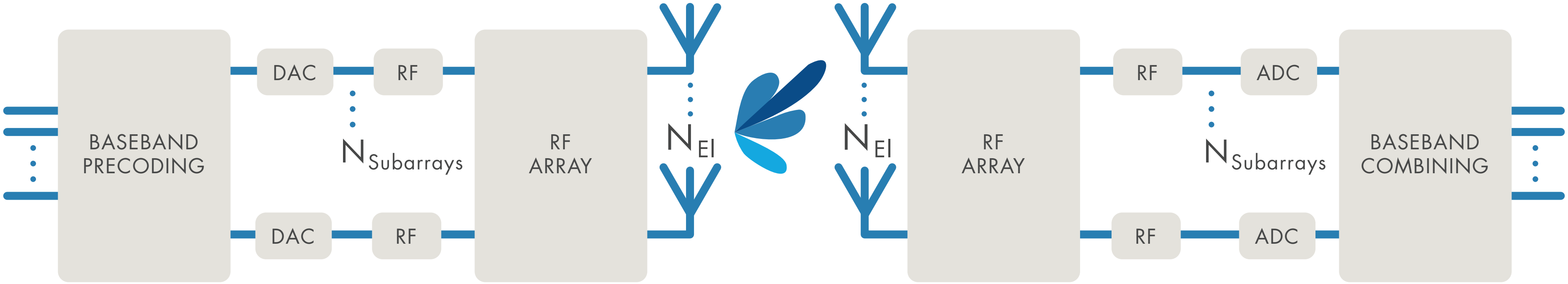
Frequency 2400 MHz

The Antenna Designer app, which provides interactive tools for antenna selection, design, and analysis.

Hybrid Beamforming

While smaller wavelengths enable massive MIMO implementation within small form factors, signal path and propagation challenges associated with mmWave frequencies also increase. To achieve better beamforming control and flexibility, it would be ideal to have independent weighting control over each antenna array element, with a transmit/receive (T/R) module dedicated to each 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.



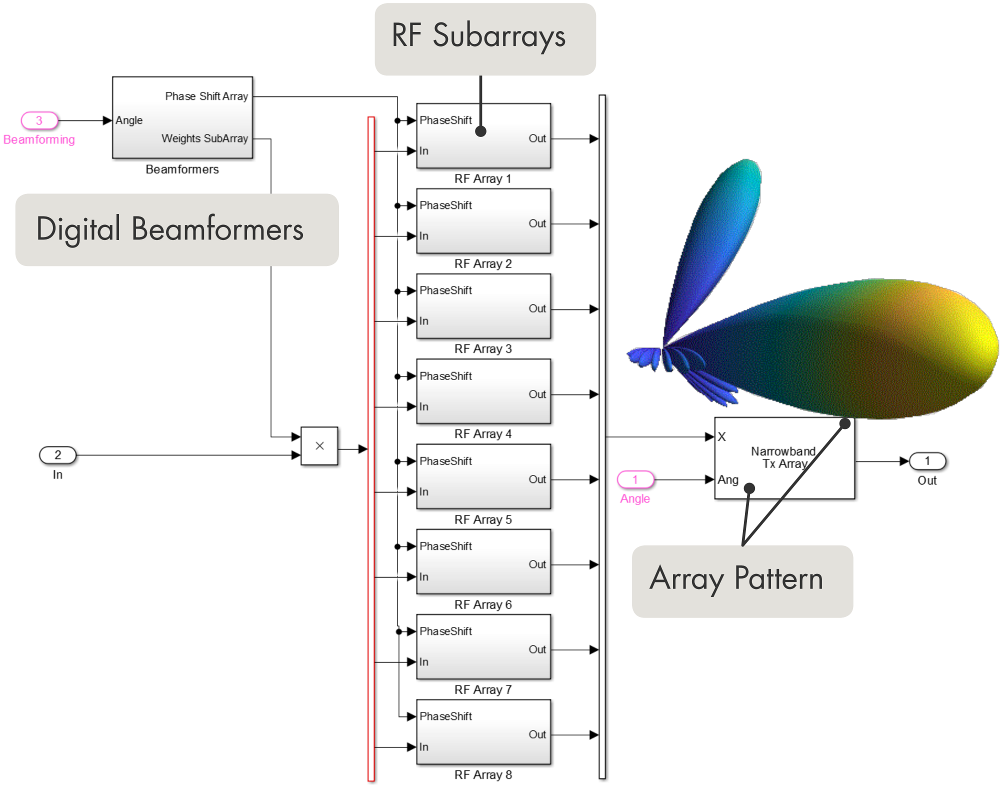
Hybrid beamforming architecture.

Hybrid Beamforming *continued*

A key challenge in hybrid beamforming design is to meet the required performance parameters while meeting 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.

The figure 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.



Hybrid beamforming design in MATLAB.

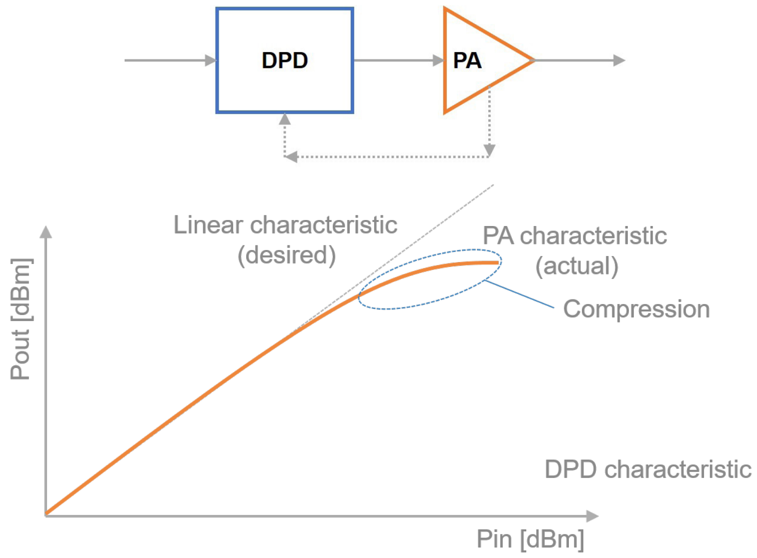
For a detailed discussion, see the white paper *Hybrid Beamforming for Massive MIMO Phased Array Systems*. The white paper uses a 64x64 element, 66 GHz millimeter wave example and demonstrates antenna array modeling and partitioning of beamforming between the digital and RF domains.

Modeling and Linearizing Power Amplifiers

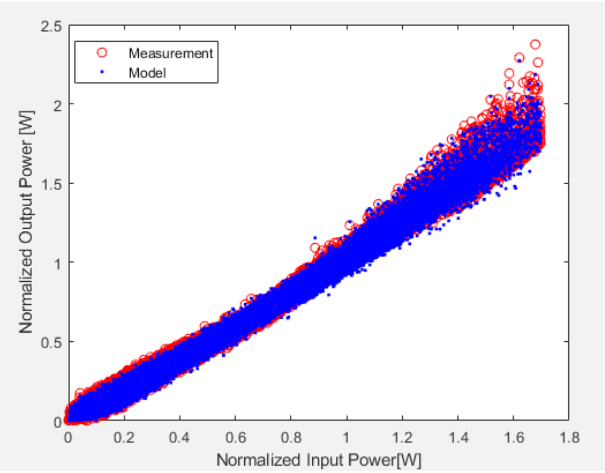
The linearity of power amplifiers (PAs) is the critical specification of every transmitter. Backing off power amplifiers to operate in the highly linear region is simply not a viable commercial solution, especially at higher frequencies and for larger bandwidths. For this reason, digital predistortion (DPD) techniques are applied to increase the efficiency of the transmitter and at the same time limit spectral regrowth and interchannel interference.

Developing a quality DPD algorithm is challenging, because it requires a deep understanding of the effects introduced by the power amplifier and adjacent subsystems, such as the antenna. Because power amplifiers are nonlinear and are affected by finite memory, the characterization of power amplifiers strongly depends on the type of signal used to drive it. For example, the behavior of the PA depends on the bandwidth of the driving signal, on its spectral occupation, and on its peak-to-average-power (PAPR) ratio.

Because of this complexity, DPD algorithms are often developed in the lab, using rapid prototyping platforms that enable you to test algorithms together with the actual PA. While this approach is useful to validate and fine-tune the algorithms, it is harder to apply when the actual PA is not yet available, or to explore the algorithmic DPD design space.



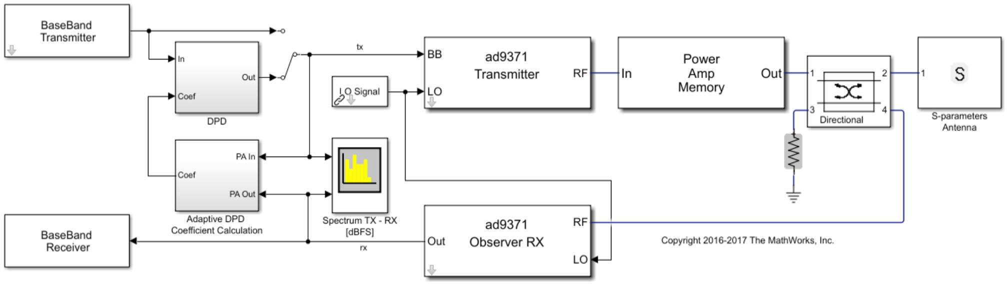
Conceptual representation of amplifier linearization using digital predistortion.



Example of a memory polynomial model of a Power Amplifier identified with MATLAB using measurement data, and simulated in RF Blockset using Circuit Envelope.

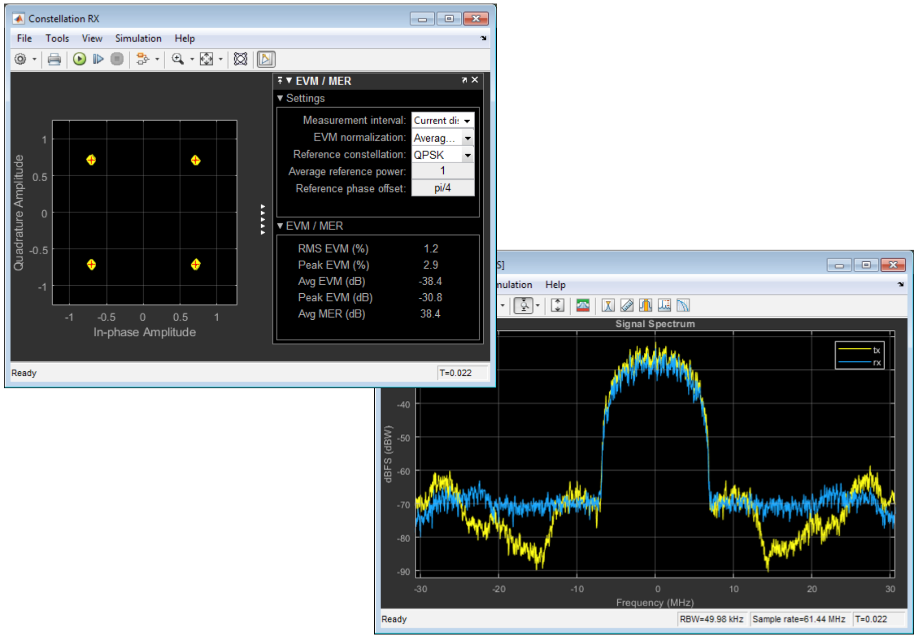
Modeling and Linearizing Power Amplifiers *continued*

For these reasons, a model-based approach is recommended to explore and develop DPD algorithms before lab prototyping and testing. However, this is only possible when a good quality model of a power amplifier is available. RF Blockset provides models of power amplifiers at different levels of abstraction, including generalized memory polynomial models that are a convenient derivation of Volterra series. The behavioral models are identified using power amplifier input/output characteristics, coming from circuit-level simulation done with IC design tools or from actual measurements. Designers can leverage the provided identification routing or use their own procedure to compute the series coefficients. In a matter of seconds, users can experiment using different polynomial orders and memory depth.



Closed-loop transceiver model with power amplifier and adaptive DPD algorithm. The lab-validated AD9371 models include real-life effects. The Volterra series model of the power amplifier includes non-linearity and memory effects. The loop simulation includes low and high-power effects, timing, and frequency selectivity over the signal bandwidth. The adaptive DPD algorithm improves the device linearity within the signal bandwidth.

Once the model is identified, it can be used within a system simulation environment together with realistic (and standard-compliant) baseband signals, models for the low-power RF transmitter and observer receiver, the antenna termination expressed with S-parameters, and different types of adaptive DPD algorithms. With this approach, designers can innovate more rapidly and validate new ideas while taking into account dispersive and nonlinear effects that are otherwise hardly reproducible and understood in the lab.



Measurement interval: Current di:
EVM normalization: Averag...
Reference constellation: QPSK
Average reference power: 1
Reference phase offset: pi/4

EVM / MER

RMS EVM (%)	1.2
Peak EVM (%)	2.9
Avg EVM (dB)	-38.4
Peak EVM (dB)	-30.8
Avg MER (dB)	38.4

RBW=49.98 kHz Sample rate=61.44 MHz T=0.022

Plots showing spectrum analysis (bottom right) and received constellation (top left).

Learn More

Ready for a deeper dive? Explore these resources to learn more about new architectures and algorithms for 5G systems.

Watch

What Is 5G Toolbox? (2:06)

Spatial Multiplexing and Hybrid Beamforming for 5G Wireless Communications (28:35)

Modeling RF Power Amplifiers and Increasing Wireless Transmitter Linearity with DPD Using MATLAB (25:11)

Explore

5G Wireless Technology Development

Beamforming

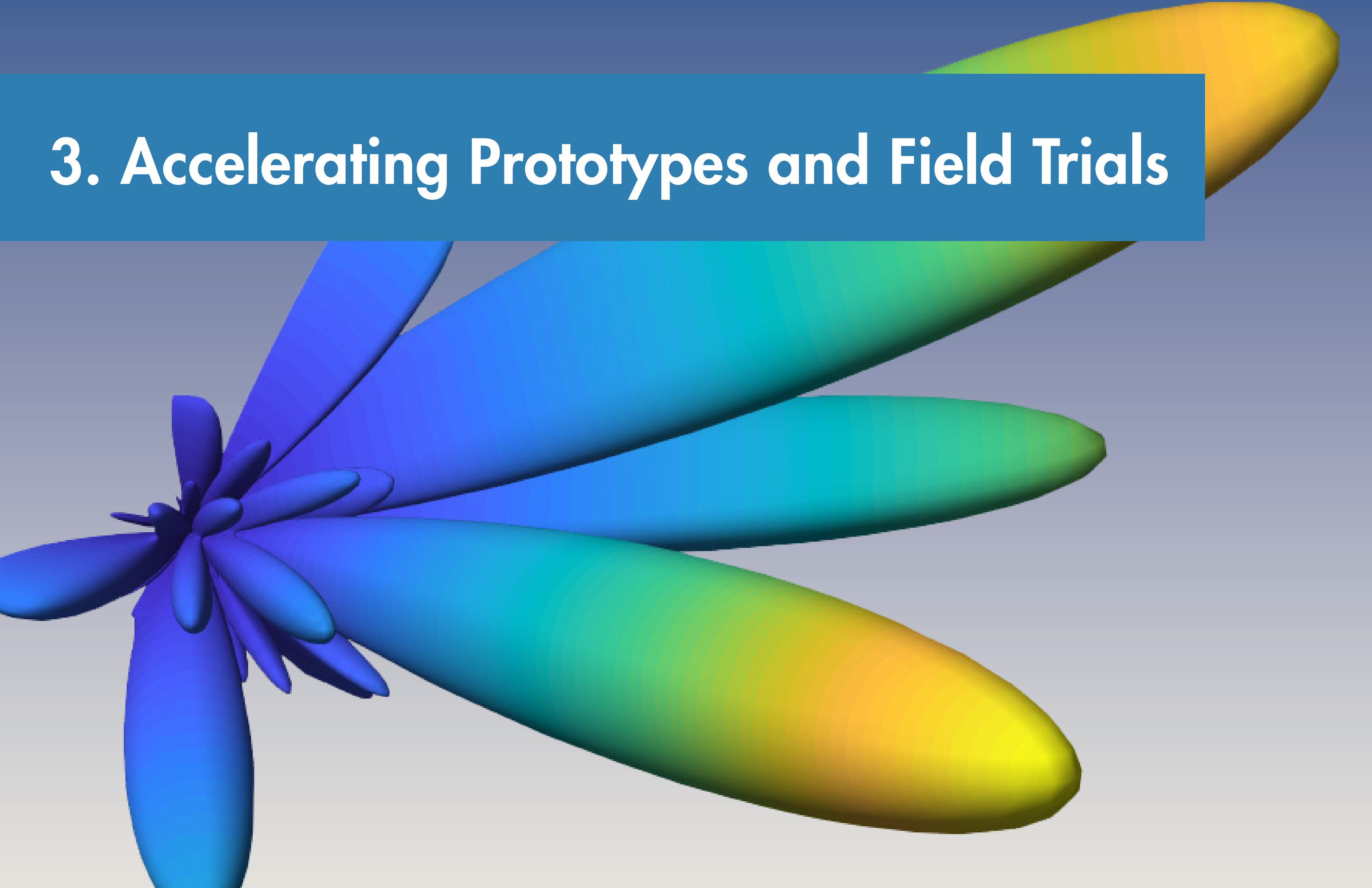
RF Systems

Channel Modeling

Massive MIMO

Developing a Radio Frequency System for Wireless at Huawei

3. Accelerating Prototypes and Field Trials

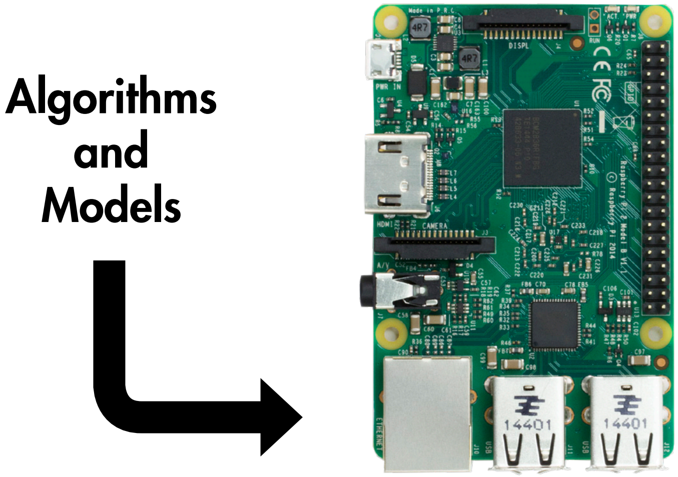


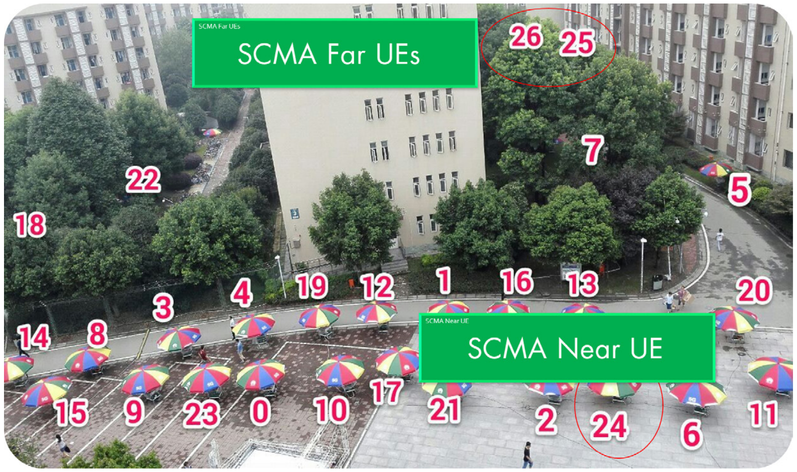
Accelerating Prototypes and 5G Field Trials

To evaluate the performance of new 5G algorithms and architectures, engineers need to develop proof-of-concept prototypes and prepare the new designs for field trials. Typically, the prototypes are built using FPGA hardware with embedded processors for some portions of the designs. These platforms are commonly known as hardware testbeds, and they enable rapid prototyping and testing of new technologies and design changes in the field.

It can be difficult for a typical R&D team to implement FPGA-based radio prototypes and testbeds without outside assistance. R&D engineers have strong signal processing and communications algorithm development expertise, but relatively little experience with hardware implementation. This experience gap is compounded by a tool and workflow gap. While R&D engineers typically use a high-level language like *MATLAB*, hardware engineers use their own design tools and hardware description languages (HDLs).

Leading companies instead have turned to Model-Based Design with MATLAB and *Simulink* to bridge these gaps and enable R&D engineers to quickly move new 5G algorithms and design changes to their FPGA-based test platforms.





Rapid prototyping of new algorithms on FPGA-based hardware for 5G field trials. (Image courtesy of Huawei)

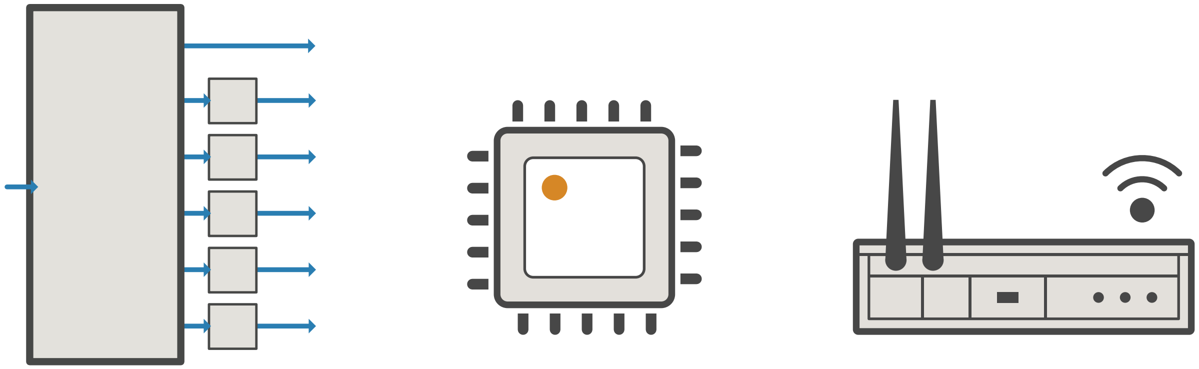
Rapid Prototyping with Model-Based Design

Many wireless engineers rely on MATLAB to develop and simulate physical layer communication algorithms. Model-Based Design with MATLAB and Simulink enables wireless engineers to use a common model for both simulation and implementation on the FPGA or SoC (system-on-chip) prototyping platform. They no longer need to maintain different representations of the same design, and they can use MATLAB code as a reference and test bench to verify the hardware design.

Using Model-Based Design, the R&D and system engineers can develop and debug hardware prototypes of their algorithms without hardware specialists. They can quickly update prototypes in response to changes in requirements. Instead of handing the hardware team an abstract algorithm to interpret and implement, they can hand off an efficient, fully verified VHDL® or Verilog® implementation.

With Model-Based Design, 5G R&D teams can:

- Explore algorithms and develop a standard-compliant reference and test bench in MATLAB
- Develop a hardware-accurate model in Simulink using fixed-point algorithm blocks
- Partition the model into subsystems to target the FPGA fabric and processor on the prototype
- Automatically generate target-independent or target-optimized HDL and C code
- Automate prototyping on SDRs and other FPGA or SoC hardware
- Integrate and test the generated code into the full radio platform design

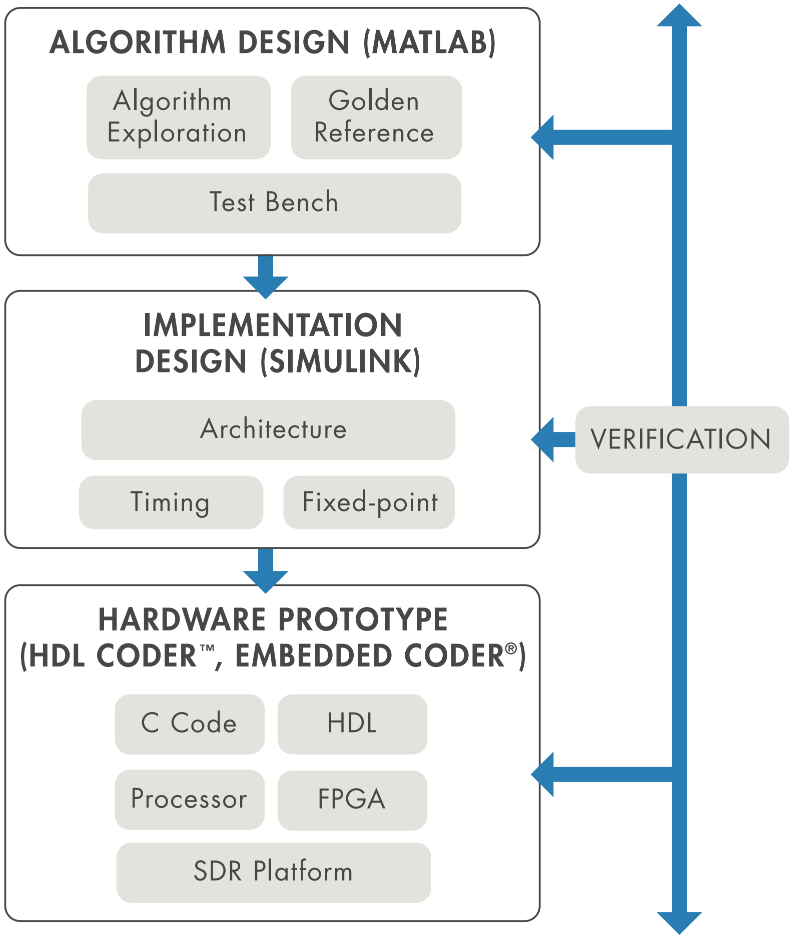


Rapid Prototyping with Model-Based Design *continued*

This process produces fully verified algorithm implementations, running C and HDL code automatically generated from a Simulink model. Available support packages for commercially available software-defined radio (SDR) frameworks further automate integration into a real-time radio application that transmits and receives signals.

To deploy the verified algorithms onto a custom hardware testbed, you can integrate the generated code with other custom interfaces and external IP cores using standard FPGA development tools.

When changes are required, you can quickly modify the algorithm model, verify it in simulation, regenerate the code (typically within minutes), integrate the code in the FPGA development environment, and synthesize it for FPGA implementation. The generated code is well structured, readable, and functionally accurate.



Using Model-Based Design for rapid prototyping of 5G algorithms.

Case Study: Developing an LTE OFDM Modulator and Detector

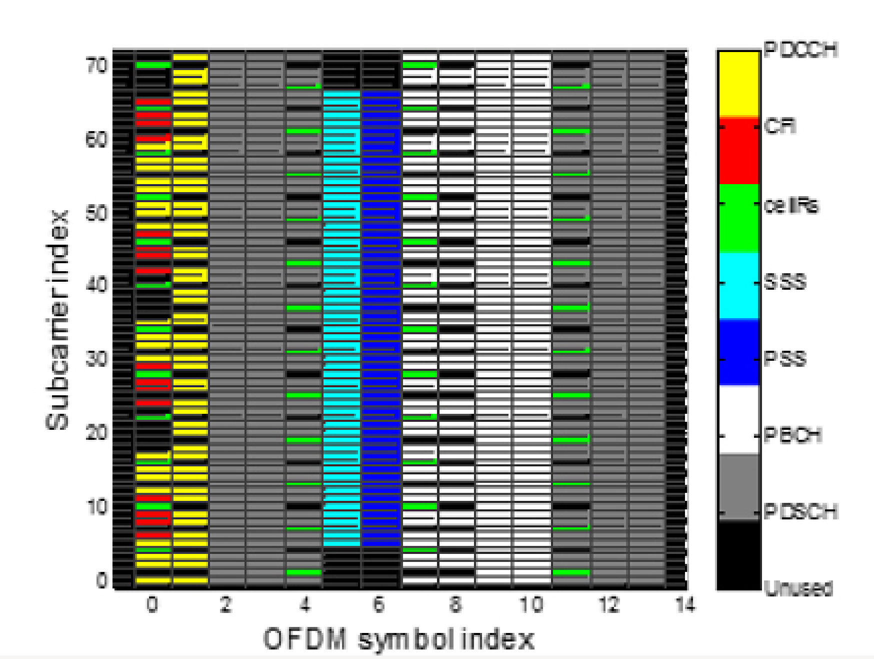
This example illustrates the application of Model-Based Design in the context of an LTE-compliant OFDM modulator and detector. The design requires a peak detection algorithm that will decode an LTE-compliant signal and execute efficiently in hardware.

The process starts with MATLAB and LTE System Toolbox to explore algorithm options to produce a standard-compliant reference waveform and test bench.

Visualizing the LTE resource grid, as shown here, helps to verify that the waveform is constructed properly and conforms to the standard.

The following image shows the steps in transforming an LTE cell search detection algorithm into HDL code ready for FPGA prototyping or implementation:

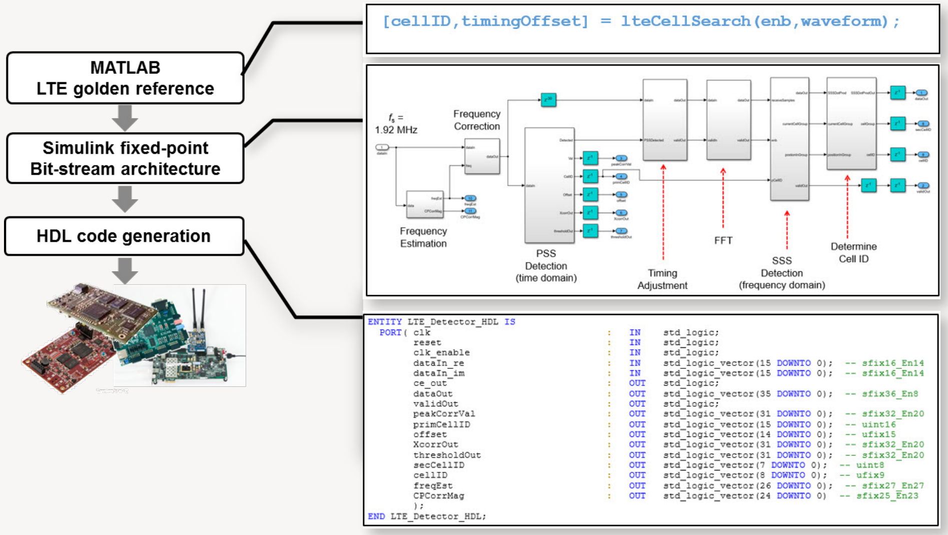
- Behavioral golden reference in MATLAB
- Hardware-accurate model in Simulink
- HDL code automatically generated from the Simulink model



LTE resource grid representing a standard-compliant waveform.

Case Study: Developing an LTE OFDM Modulator and Detector *continued*

The detector performs frequency estimation and correction, PSS detection, timing adjustment, FFT, and SSS detection to determine the LTE cell identity of a detected cell group. The model was verified using an LTE-compliant waveform, and it successfully detects cell IDs both in simulation and over-the-air running on a Xilinx® Zynq® SDR.

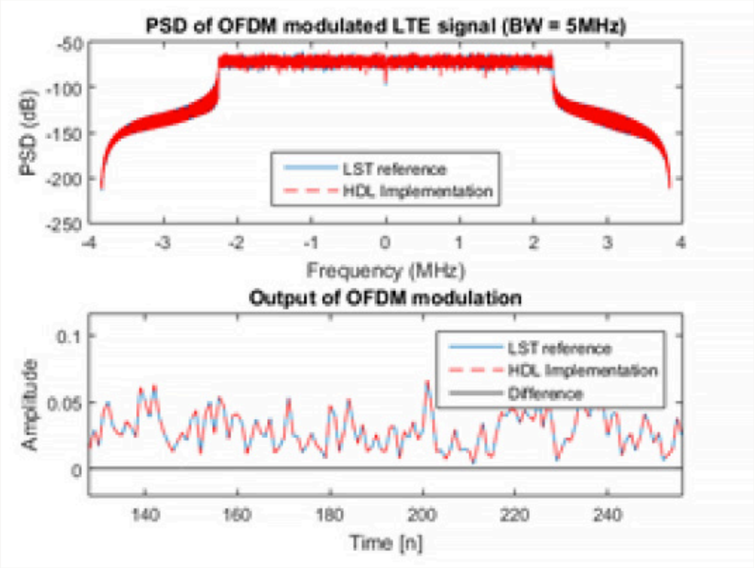


```
[cellID,timingOffset] = lteCellSearch(enb,waveform);
```

```
ENTITY LTE_Detector_HDL IS
  PORT( clk                              :   IN    std_logic;
        reset                            :   IN    std_logic;
        clk_enable                       :   IN    std_logic;
        dataIn_re                        :   IN    std_logic_vector(15 DOWNTO 0);  -- sfix16_En14
        dataIn_im                        :   IN    std_logic_vector(15 DOWNTO 0);  -- sfix16_En14
        ce_out                           :   OUT   std_logic;
        dataOut                          :   OUT   std_logic_vector(35 DOWNTO 0);  -- sfix36_En8
        validOut                         :   OUT   std_logic;
        peakCorrVal                      :   OUT   std_logic_vector(31 DOWNTO 0);  -- sfix32_En20
        primCellID                       :   OUT   std_logic_vector(15 DOWNTO 0);  -- uint16
        offset                           :   OUT   std_logic_vector(14 DOWNTO 0);  -- ufix15
        XcorrOut                         :   OUT   std_logic_vector(31 DOWNTO 0);  -- sfix32_En20
        thresholdOut                     :   OUT   std_logic_vector(31 DOWNTO 0);  -- sfix32_En20
        secCellID                        :   OUT   std_logic_vector(7 DOWNTO 0);  -- uint8
        cellID                           :   OUT   std_logic_vector(8 DOWNTO 0);  -- ufix9
        freqEst                          :   OUT   std_logic_vector(26 DOWNTO 0);  -- sfix27_En27
        CPCorrMag                        :   OUT   std_logic_vector(24 DOWNTO 0)  -- sfix25_En23
        );
END LTE_Detector_HDL;
```

MATLAB reference code for LTE cell search (top), and workflow for designing and generating an HDL implementation of the algorithm (bottom).

The output of simulations and hardware tests can be compared with the reference algorithm and visualized in MATLAB, as shown below.



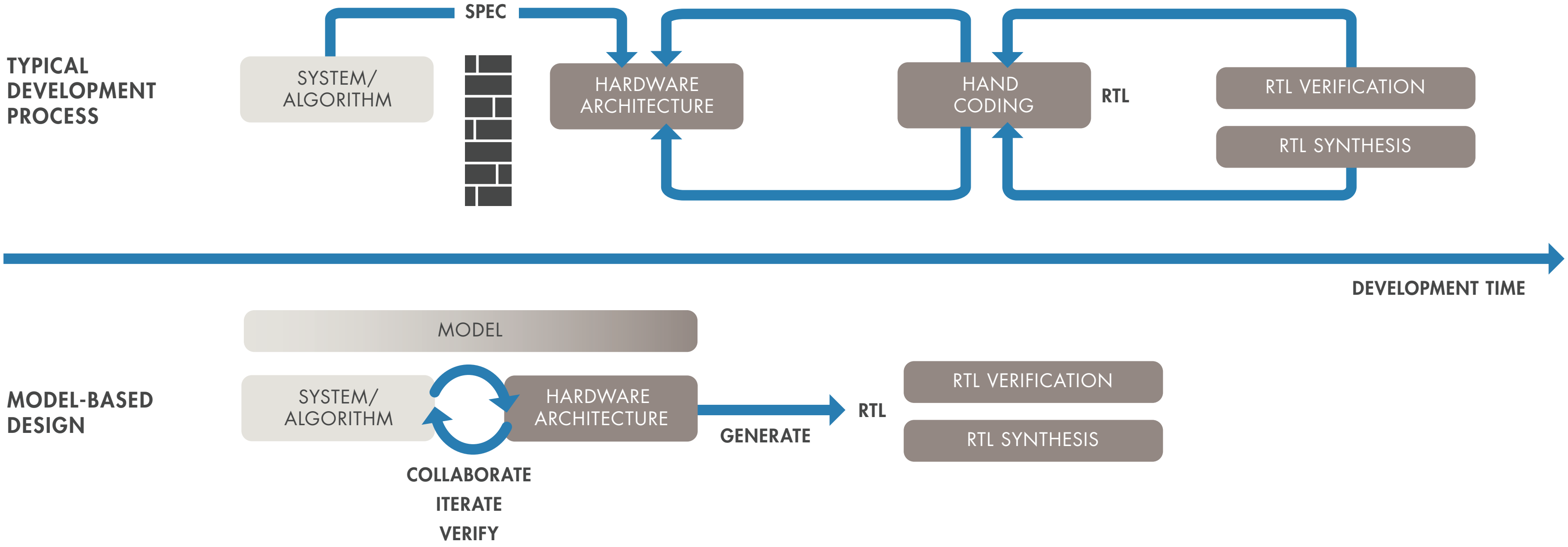
Verification of the HDL implementation of the OFDM modulation.

To learn more, download an example of *building an LTE-compliant OFDM modulator and detector* for implementation with *HDL Coder™*, and use *LTE System Toolbox* to verify the HDL implementation model.

HDL Implementation on FPGA and ASIC

Model-Based Design doesn't stop at prototyping. Wireless engineers are successfully using 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 implementations of algorithms for multirate filtering, PAPR suppression, digital predistortion, and baseband processing.

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, verifying the results in Simulink, and regenerating the HDL code. HDL Coder integrates with SoC and FPGA design to provide target-optimized implementations. This can accelerate the development of SoC and FPGA designs, enabling teams to complete this work in days or weeks rather than in months.



Using Model-Based Design to accelerate algorithm implementation on FGPAs and ASICs.

Who uses this workflow for 5G prototyping? *continued*

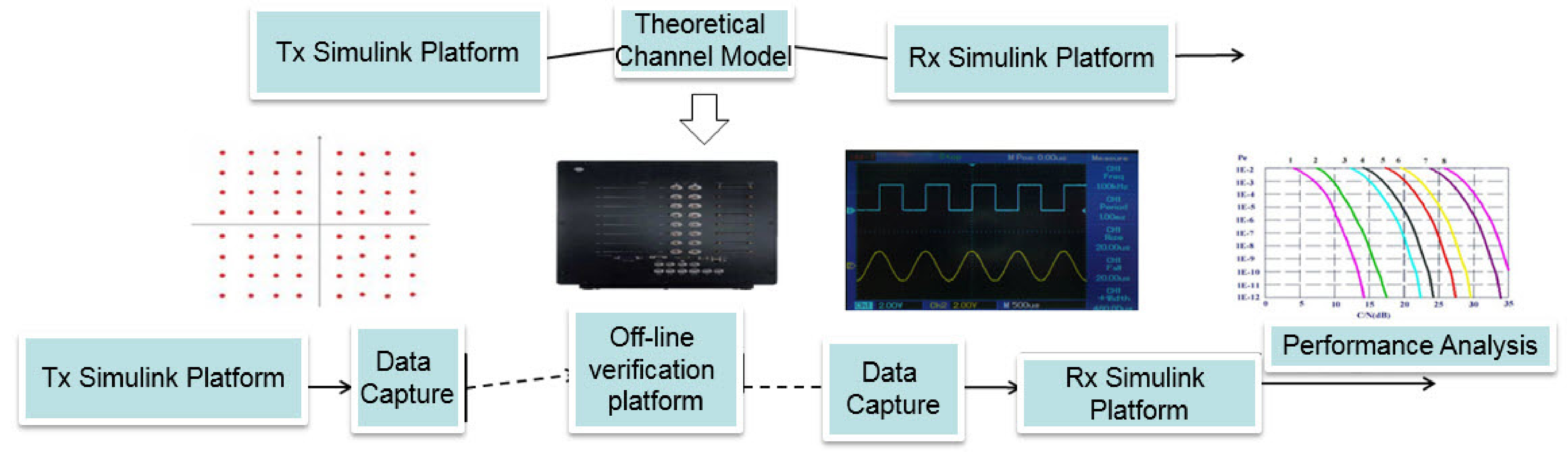
Huawei: Accelerating 5G Prototyping

Kevin Law, director of algorithm architecture and design at Huawei, provides an insider view of a 5G development workflow using MATLAB and Simulink to accelerate 5G prototyping.

For details on how Huawei engineers use MATLAB and Simulink in their 5G R&D, see the user story:

5G R&D at Huawei: An Insider Look—Accelerating the move from theory to engineering practice with MATLAB and Simulink

> "When we do 5G prototyping design and development, we automatically generate HDL code from MATLAB for hardware verification. When we evaluate the algorithm in field testing, we can very effectively see the whole system's performance and status, and identify any potential issues."
>
> — Kevin Law, Director of Algorithm Architecture and Design, Huawei



Ultra-high-speed algorithm simulation and offline verification using Simulink.

Who uses this workflow for 5G prototyping? *continued*

Ericsson: Deploying 5G Hardware Testbeds

To complete and deploy a full proof-of-concept on a hardware testbed, a development group at Ericsson has integrated algorithms developed with MATLAB and Simulink with other system components in a standard FPGA development environment. Specifically, they integrated HDL code automatically generated from Simulink using HDL Coder with manually coded custom interfaces and IP cores from FPGA vendors for components such as soft processors, DMA, and physical interfaces. When changes are required, the engineers make the change in Simulink, regenerate the code (typically within minutes), integrate the code in the FPGA development environment, and synthesize it for FPGA implementation. Ericsson engineers have noted that the generated code is well structured, readable, and functionally accurate.

For details on how Ericsson is using hardware testbeds for fast hardware prototyping and automatic HDL code generation with MATLAB and Simulink, watch the video *Radio Testbed Design Using HDL Coder* (22:44).

> "When a new design is required because a new design idea emerges, using HDL Coder, we were able to demonstrate that the change was acceptable and have a working demonstration in less than one week."
>
> — Tomas Andersson, Ericsson

Learn More

Ready for a deeper dive? Explore these resources to learn more about accelerating 5G prototypes and field trials.

Qualcomm: Connecting Systems and HDL World – Rapid RTL Generation (Conference Proceedings)

Huawei: System-Level ASIC Algorithm Simulation Platform Using Simulink (Conference Proceedings)

Watch

An FPGA Implementation of an LTE Design (41:53)

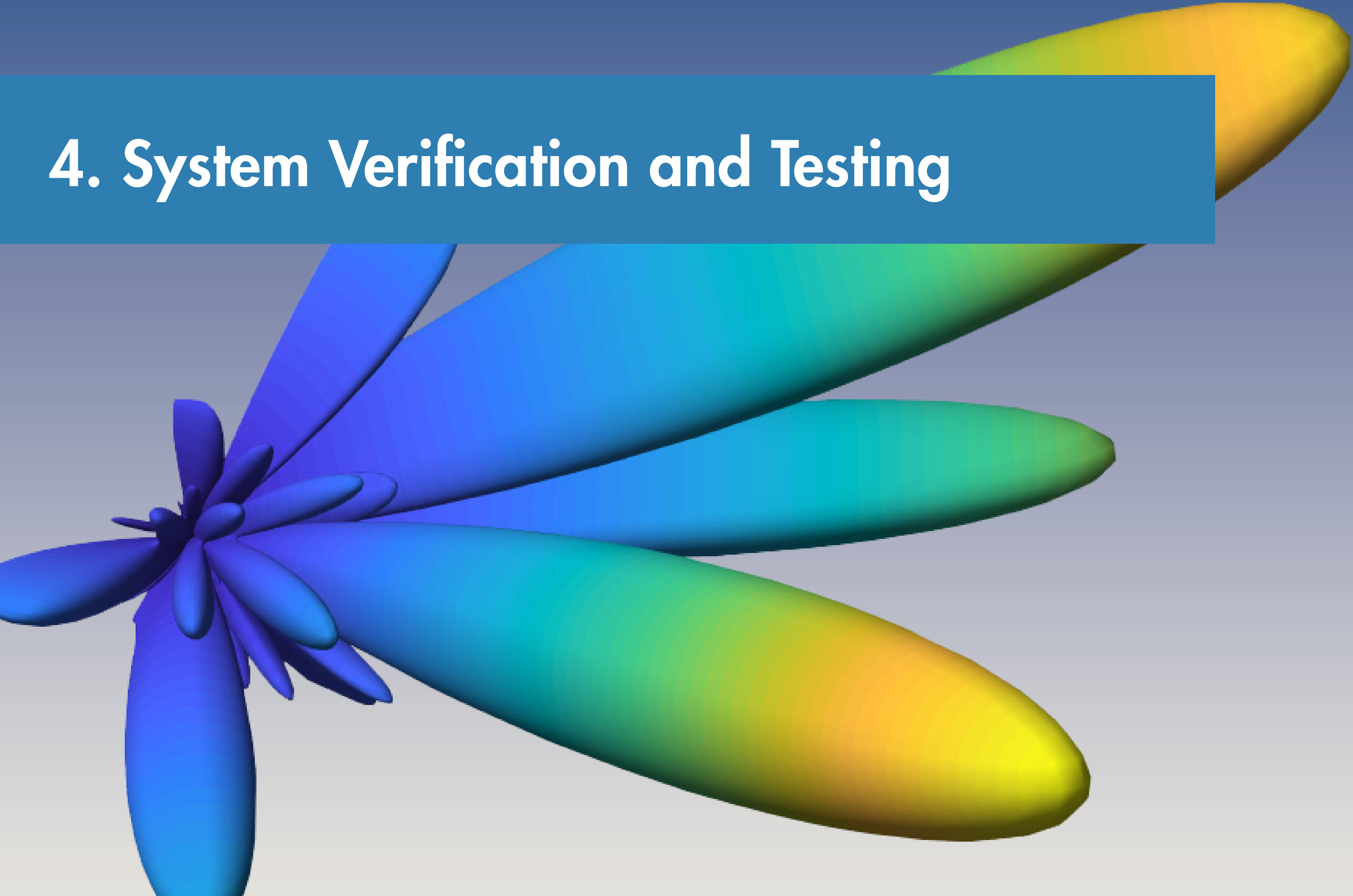
Download Code

HDL Implementation of LTE OFDM Modulator and Detector

Explore

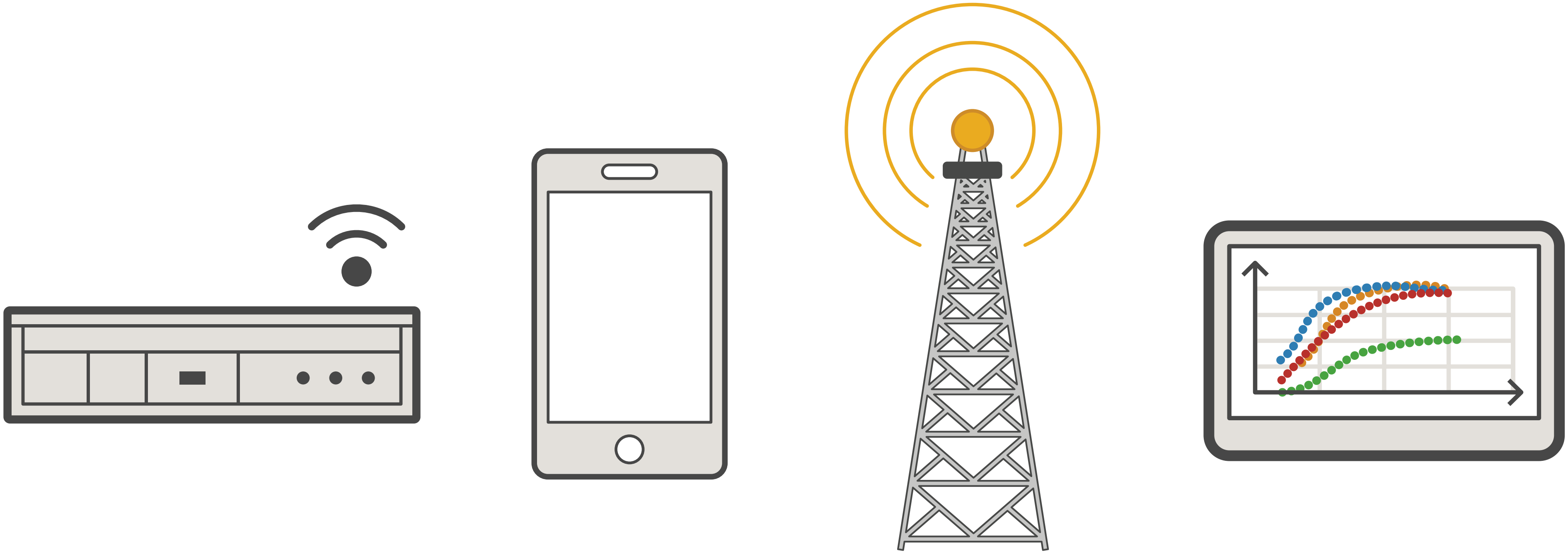
HDL Code Generation and Verification

4. System Verification and Testing



5G System Verification and Testing

After completing baseband, RF, and antenna array designs, it is essential to verify whether the system designs yield satisfactory results in the lab and in the field under real-world conditions. To do so, engineers need to work with live 5G signals and waveforms using interfaces to test and measurement equipment.



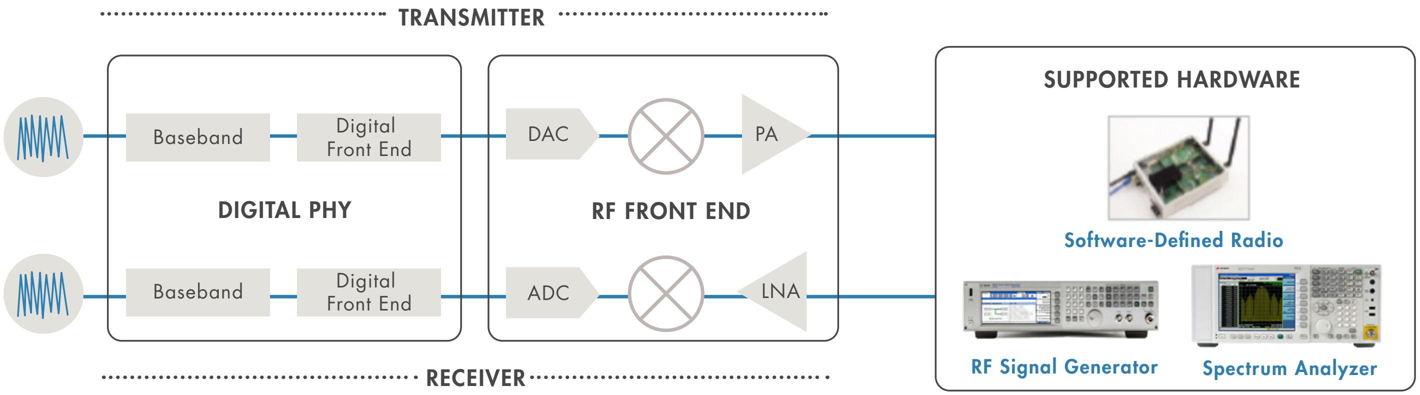
5G Live-Signal Generation and Reception

By connecting to software-defined radio (SDR) and RF instrument hardware, you can use *MATLAB®* and *Simulink®* to perform over the-air tests to validate your 5G designs in simulation, in the lab, or in the field 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 and reduces test development time and errors.

You can capture live 5G or LTE signals for analysis and comparison to baseband simulation results. Algorithm designers can use the captured signals to test their algorithms, and the RF team can use this setup to verify their RF design. *Download an example that demonstrates live LTE signal generation and capture capability.*

During field testing, it is important to be able to quickly modify test parameters and test scripts to accommodate different test scenarios. Using MATLAB provides the flexibility to customize tests and to diagnose and debug subtle issues that are difficult to address in the more constrained software environments and encrypted waveforms that test and measurement instruments typically provide.

MATLAB and Simulink support a range of available SDR hardware, as well as RF signal generators and spectrum analyzers from RF instrument vendors such as Keysight, Rohde & Schwarz, National Instruments, and Anritsu. SDR support packages are available for Xilinx Zynq and FPGA Radios; USRP® N, X, and E Series Radios; PlutoSDR; and RTL-SDR. From these options, you can choose the hardware that's most appropriate for your requirements and budget.



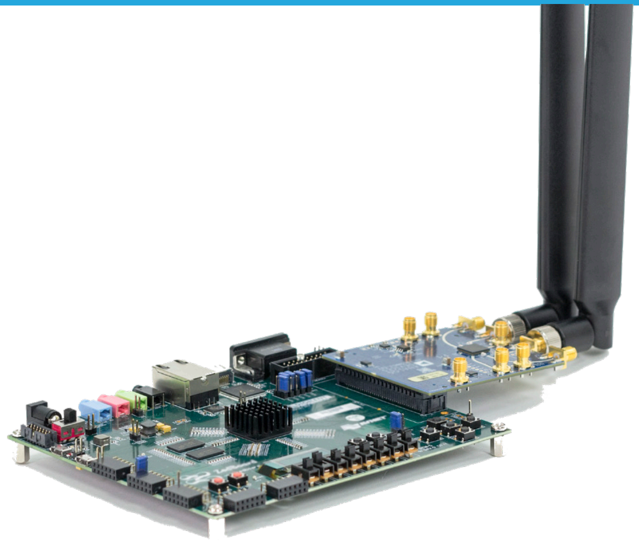
MATLAB and Simulink support for over-the-air (OTA) testing with a range of available SDR and RF instrument hardware, simplifying comparison of OTA tests to simulation results.

5G Live Signals via Software-Defined Radio

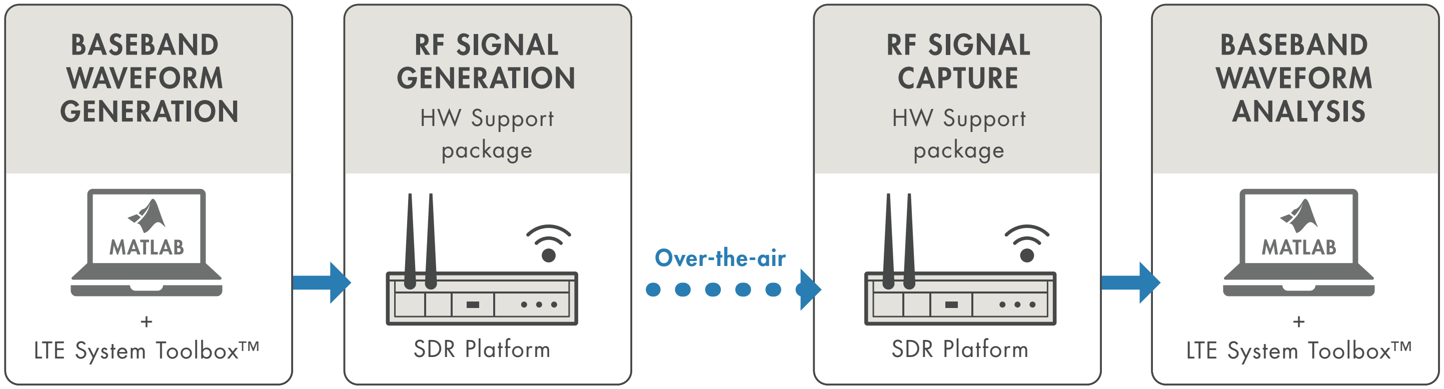
A software-defined radio (SDR) is a wireless device consisting of a configurable RF front end with an FPGA or programmable SoC to perform digital functions. Commercially available SDR hardware can operate at different frequencies to transmit and receive standard-compliant and custom signals to test baseband algorithms and RF front end designs.

You can use MATLAB and Simulink with SDR hardware as a cost-effective, real-time platform for a range of wireless engineering tasks, including:

- Over-the-air lab and field testing with live RF signals
- Signal capture for posttest analysis
- Rapid verification with custom radio functions



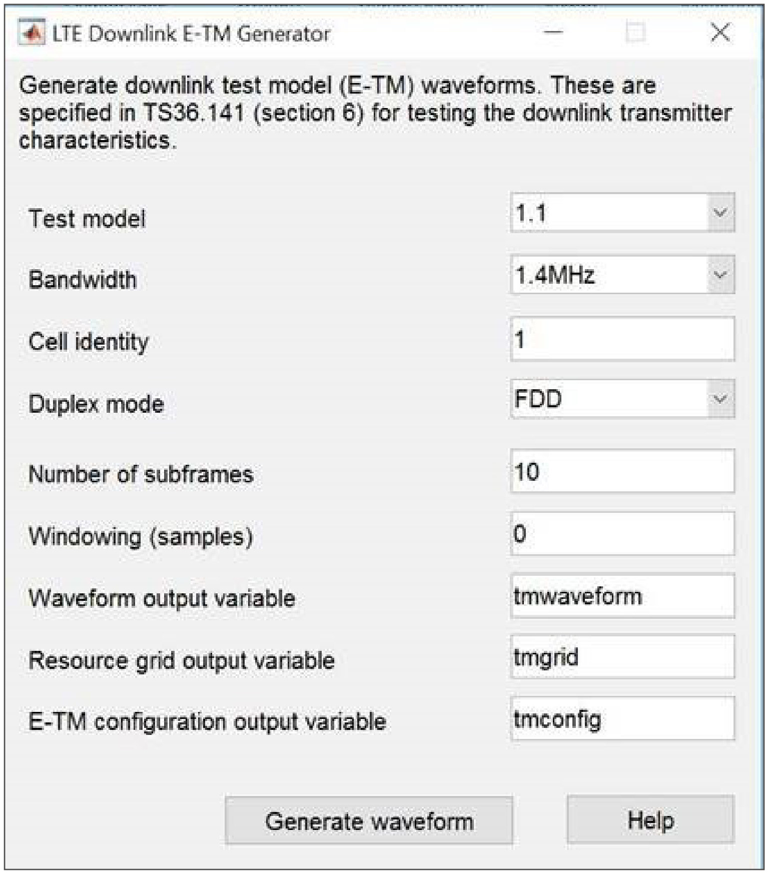
A typical SDR device with RF front end and FPGA board.



MATLAB and LTE System Toolbox with supported SDRs. This provides a flexible, cost-effective environment for live signal generation and capture.

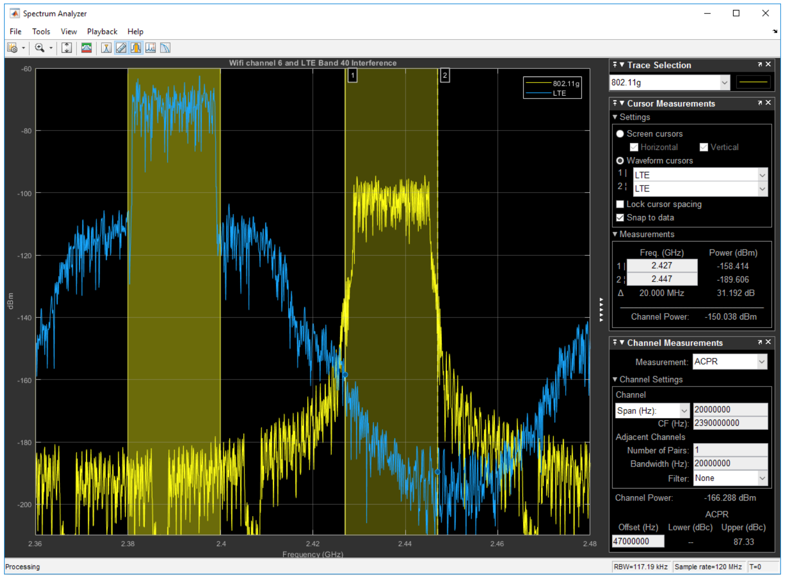
5G Live Signals via Software-Defined Radio *continued*

With an interactive app or a few lines of MATLAB code, you can adapt SDR hardware to generate baseband signals that span a wide range of possibilities—a candidate 5G signal, a 4G LTE standard-compliant signal, or a fully customized signal.



Generate downlink test model (E-TM) waveforms. These are specified in TS36.141 (section 6) for testing the downlink transmitter characteristics.

Test model	1.1
Bandwidth	1.4MHz
Cell identity	1
Duplex mode	FDD
Number of subframes	10
Windowing (samples)	0
Waveform output variable	tmwaveform
Resource grid output variable	tmgrid
E-TM configuration output variable	tmconfig



Trace Selection: 802.11g

Settings

○ Screen cursors ☑ Horizontal ☑ Vertical

● Waveform cursors

1 LTE

2 LTE

□ Lock cursor spacing

☑ Snap to data

Measurements

	Freq. (GHz)	Power (dBm)
1	2.427	-158.414
2	2.447	-189.606
Δ	20.000 MHz	31.192 dB

Channel Power: -150.038 dBm

Measurement: ACPR

Channel Settings

Channel

Span (Hz): 20000000

CF (Hz): 2390000000

Adjacent Channels

Number of Pairs: 1

Bandwidth (Hz): 20000000

Filter: None

Channel Power: -166.288 dBm

ACPR

Offset (Hz)	Lower (dBc)	Upper (dBc)
47000000	--	87.33

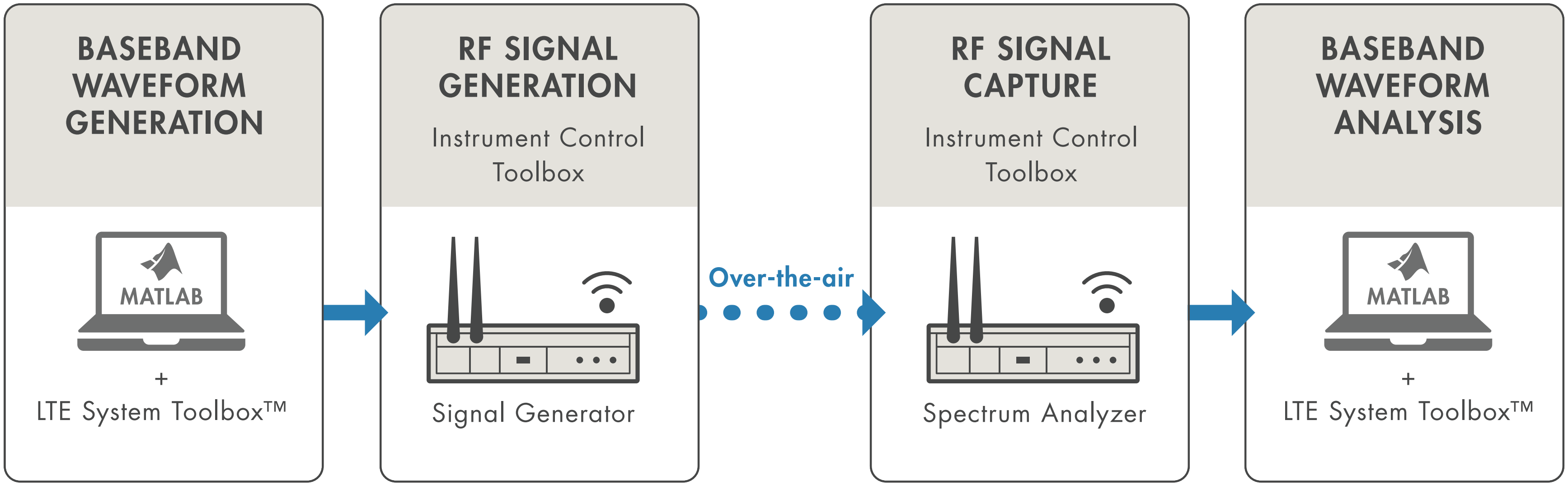
RBW=117.19 kHz | Sample rate=120 MHz | T=0

Generation (left) and analysis (right) of an LTE downlink test model (E-TM) waveform.

Generation and Capture of Live 5G Signals Using RF Instruments

The second method for working with live signals uses commercially available test instruments. These instruments employ finely calibrated RF characteristics, making them suitable for precise lab, field, or production tests. However, 5G experiments and field trials require greater software flexibility and analysis capability than is typically found on such instruments. These limitations can be overcome by using MATLAB and *Instrument Control Toolbox™*, which supports interfaces and standards such as VISA, TCP/IP, SPI, USB, and GPIB; equipment from different manufacturers; and instrument types, such as *signal generators and vector signal analyzers.*

To transmit signals, you can generate the 5G baseband signal, download it to a signal generator, configure and control the parameters in the baseband signal, and upconvert the IQ waveform to RF and play back over the air. On the receiver side, you can retrieve IQ data into a PC for analysis, perform visualization and analysis, programmatically configure acquisition parameters, and down-convert the RF signal to baseband in the hardware.



Over-the-air testing with MATLAB and test and measurement instruments.

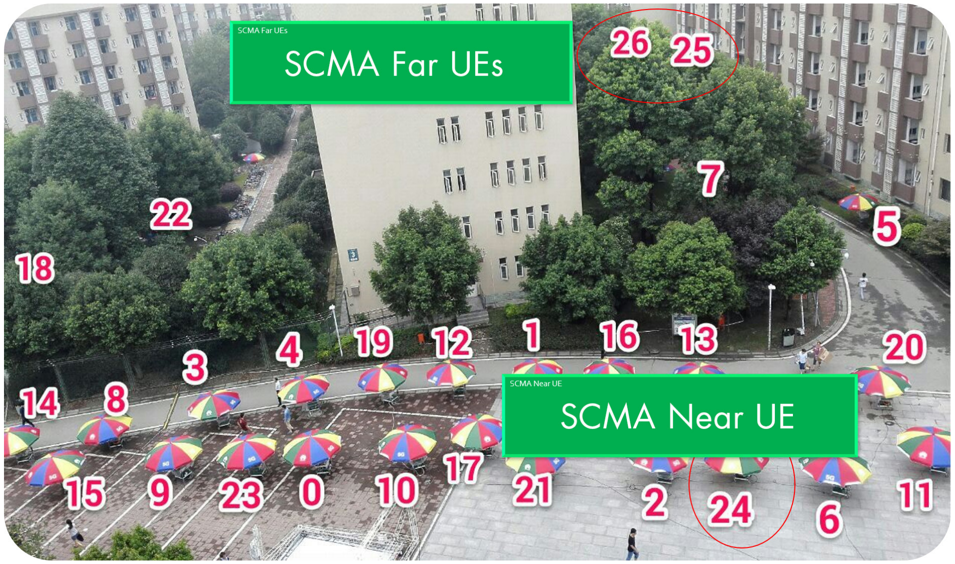
Analysis of Massive Field Trial Data Sets

Network equipment vendors and wireless operators have been conducting public field trials to demonstrate their 5G technology and characterize its operation in a real-world setting.

In 5G field trials, many parameters are measured and monitored dynamically, such as reference signal received power (RSRP) and reference signal received quality (RSRQ). System performance is recorded by capturing signals and sweeping across a large range of parameters.

After taking the measurements, designers can extract useful and actionable findings from the data. Compared with 4G systems, 5G field tests capture massive amounts of data due to high data rates and the need to capture and analyze massive MIMO radiation patterns.

The field trial system requires a test and analysis system that supports scalable data capture, data processing, analysis, and sharing of results.



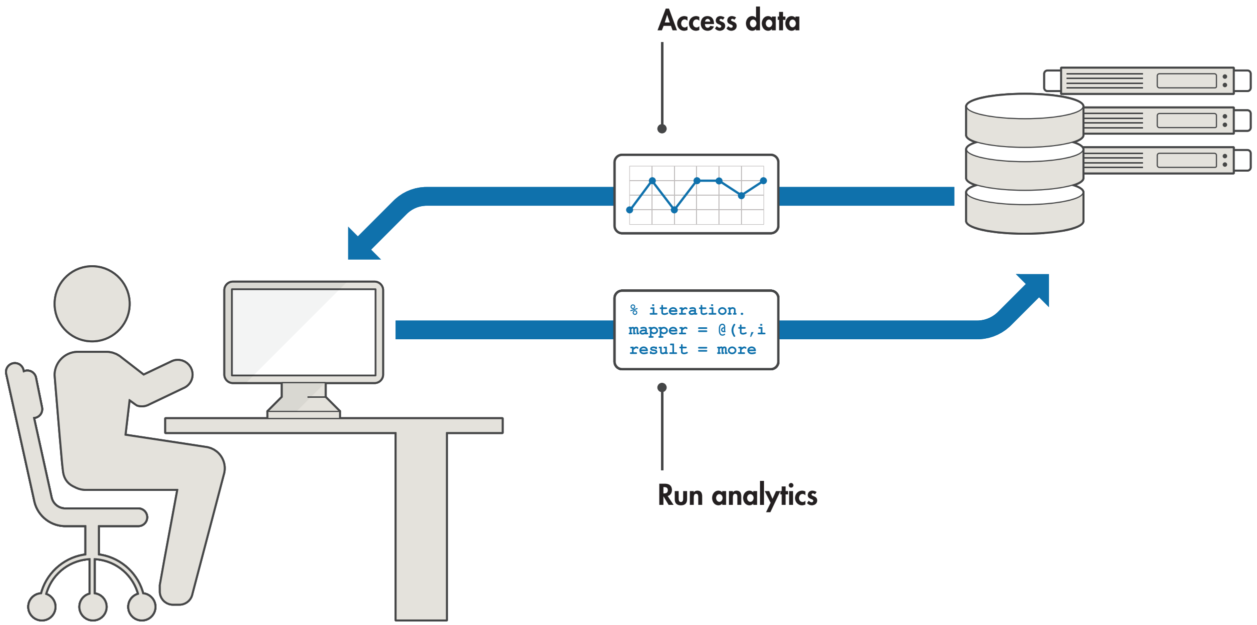
Huawei field trial from Tokyo.

Test Data Capture and Analysis

Test engineers performing field tests often use commercially available test instruments. However, 5G field trial analysis requirements go far beyond simple measurements.

Field trial analysis software must be able to import data directly from test instruments or from stored data in a variety of formats. The data represents captured signals, name and time stamps, and configuration information such as antenna pattern files. It is helpful to have a common environment or tool that can access various kinds of data from different equipment vendors.

For very large data sets, teams can use computing clusters or cloud storage that scales from a single workstation to compute clusters or process big data sets.



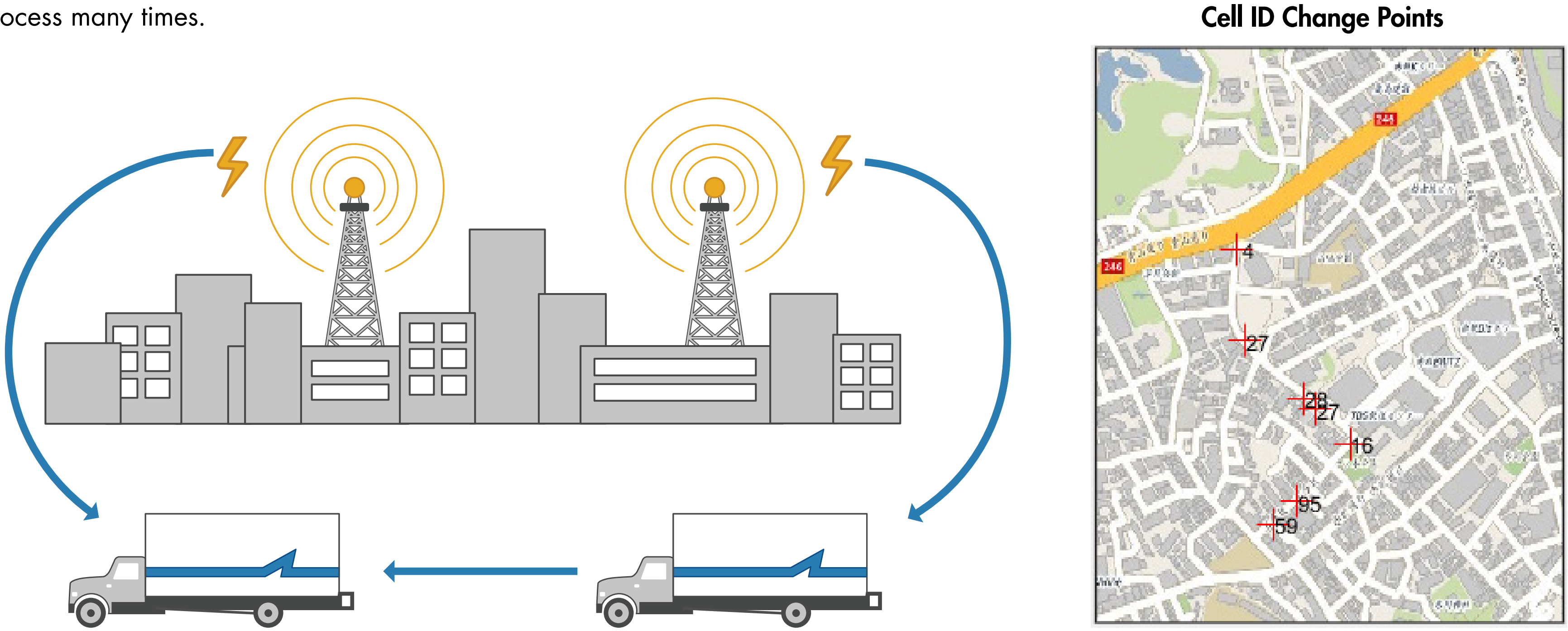
```
% iteration.
mapper = @(t,i
result = more
```

Using MATLAB to capture, process, and analyze field test data stored locally or in the cloud.

Data Processing and Visualization

Test engineers may want to store raw captured data or show the results to their management, partners, or customers after analysis.

For example, a team might want to show cell handover points on a map, decide if signal-to-interference-plus-noise ratio (SINR) is satisfactory, and how the RSRP varies. In performing this task, it can be necessary to quickly generate standard waveforms and repeat the test process many times.



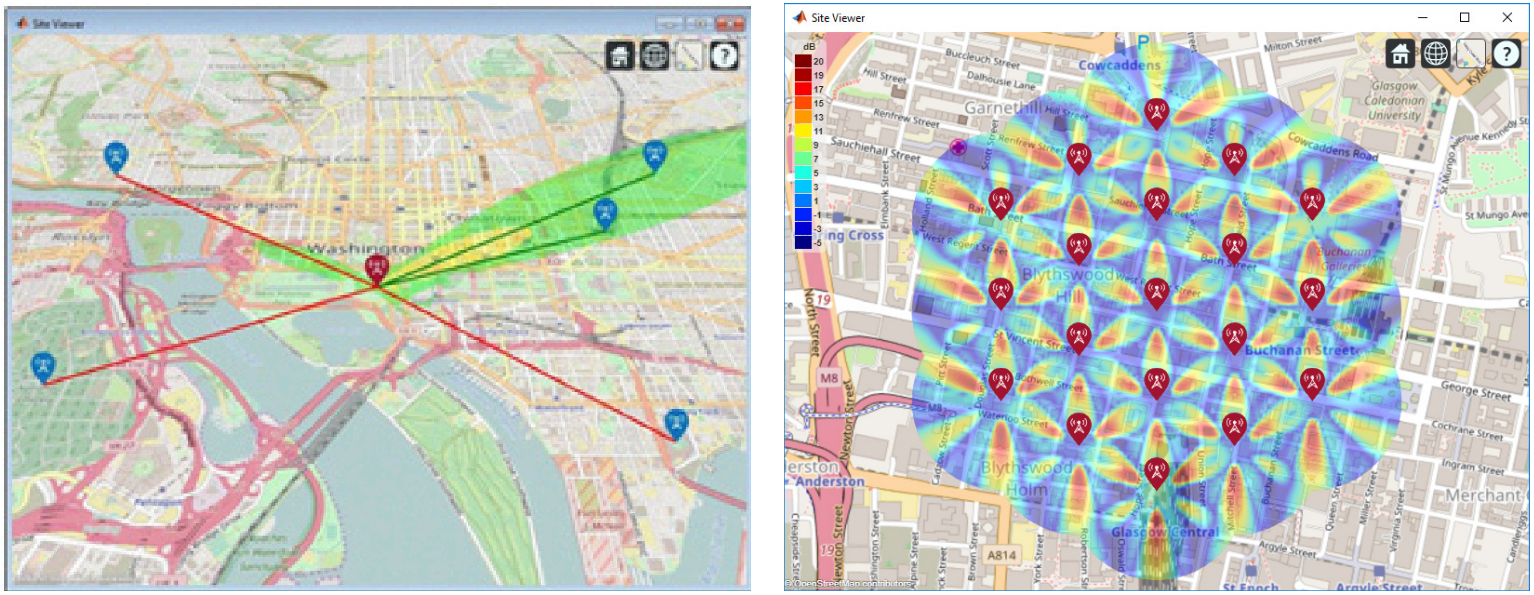
Analysis of the signal in conformity with the standard and handover point and cell ID number.

Visualization of 5G Field Test Results on Maps

Visualization is critical in field testing. Test engineers need to superimpose captured signals as well as performance and parameter data on a representation of a geographic map. 3D and pseudo-3D representations require latitude and longitude in the map.

For some 5G applications such as vehicle-to-vehicle communication and base station coverage analysis, engineers need to directly access map data with latitude, longitude, and building location information.

If properly implemented, this visualization architecture enables engineers to comprehensively visualize system performance in real-world scenarios and demonstrate results to inform network planning decisions.

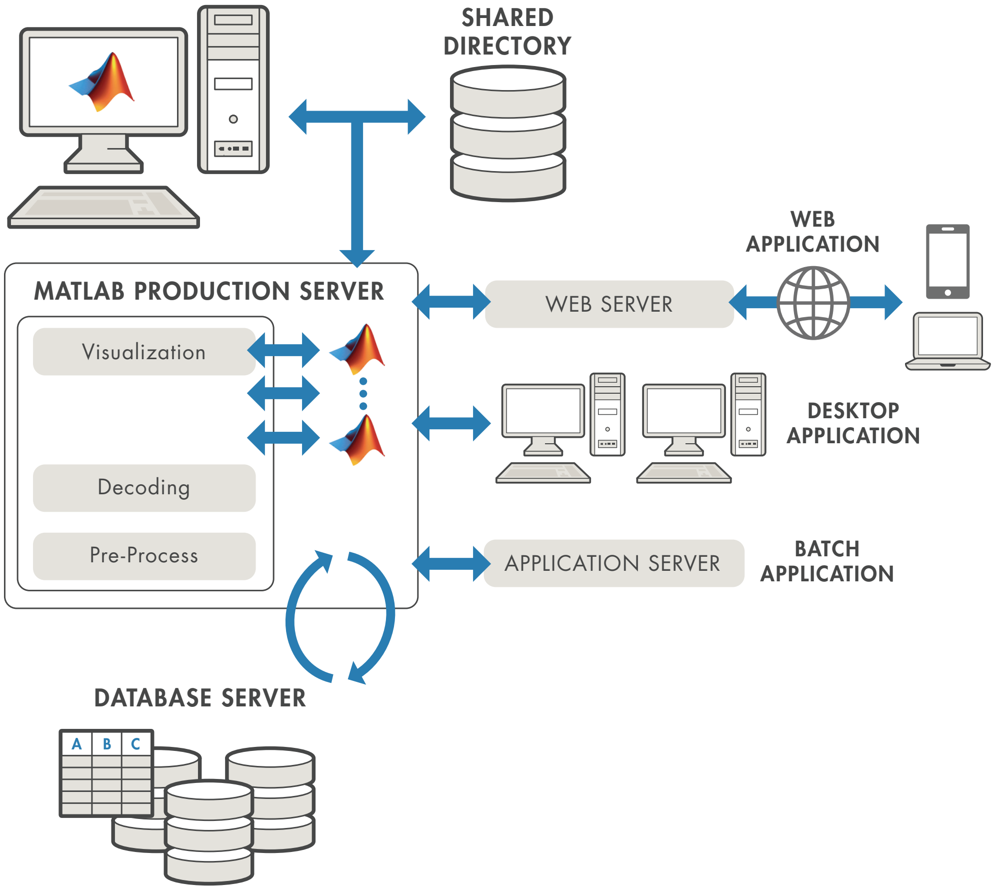


Visualization of cellular signal coverage (left) and 5G urban macro cell SINR (right) on street maps.

Analysis and Results Sharing

For large-scale postprocessing of data and central management, system architects can deploy their 5G field testing software as applications on enterprise systems. These applications can be hosted on a dedicated server or cloud, so the system architects can remotely check the test data and centrally manage system updates. Using MATLAB application deployment tools, the development team can leverage their previous work, and everyone on the team (system architects and developers, field engineers, and management team) can remotely view and validate performance and analysis results.

You can implement enterprise applications with *MATLAB Production Server™*, which lets you incorporate custom algorithms, tests, and analytics into web, database, and production enterprise applications running on dedicated servers or a cloud. See an *example of this deployment process*. This 5G field test workflow has been used by cellular service providers and leading suppliers and is ideally suited for the demands of 5G field trials.



IT architecture of a 5G field trial.

Learn More

Ready for a deeper dive? Explore these resources to learn more about 5G system verification and testing.

Watch

Wireless Standard Waveform Generation in MATLAB for Simulation and Testbeds (31:03)

Learn

Verifying LTE Designs Using Live Signals and Test and Measurement Equipment

Explore

Designing and Testing LTE Systems

© 2017 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See mathworks.com/trademarks for a list of additional trademarks.
Other product or brand names may be trademarks or registered trademarks of their respective holders.