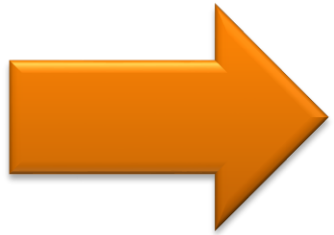
What's New in MATLAB and Simulink for Signal Processing

Jonas Rutström
Senior Application Engineer

So, what's new?

NORDIC MATLAB EXPO 2014

R2014b  *R2016a*

“What’s New in MATLAB and Simulink for Signal Processing”

Signal Processing

Audio

Antenna to Bits

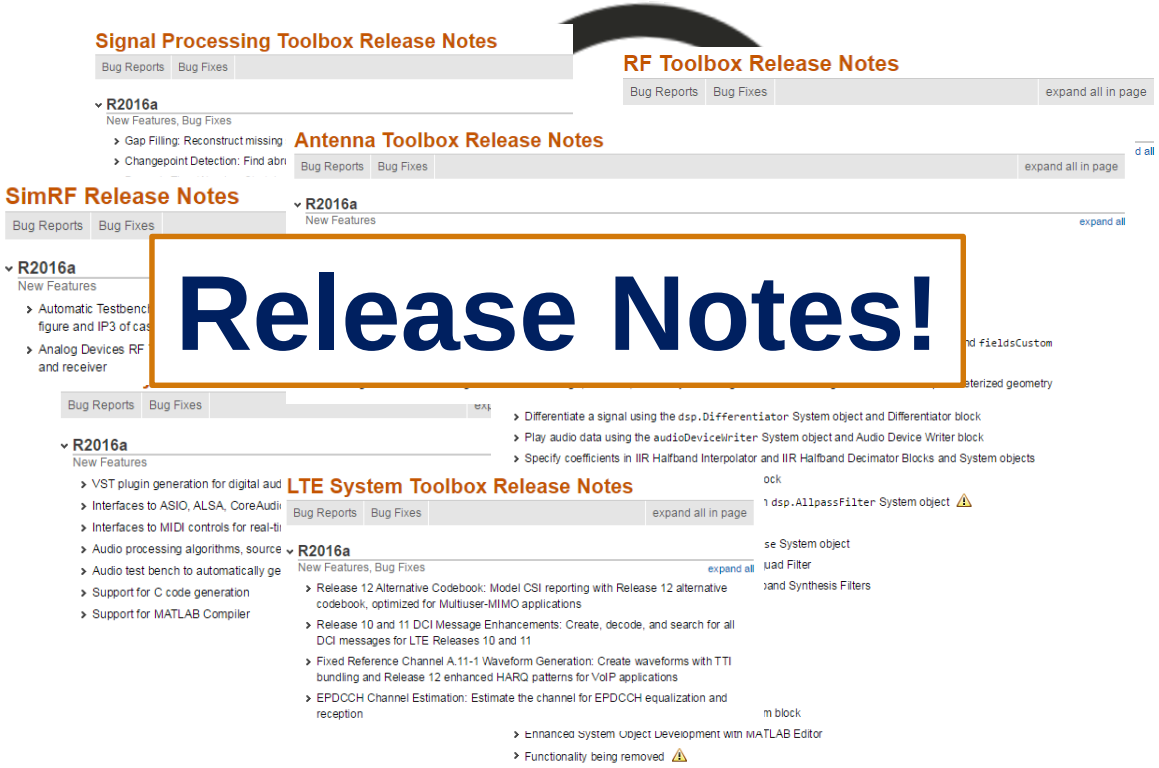
WLAN/LTE

Image and Video Processing

A few words about “What’s New?”

Details

A few words about “What’s New?”



Signal Processing

Audio

Antenna to Bits

WLAN/LTE

Image and Video Processing

Signal Processing

Audio

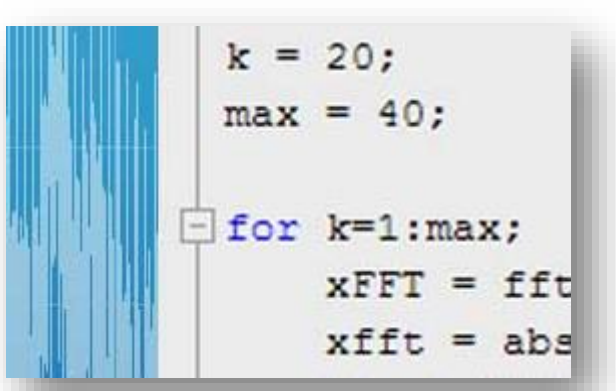
Antenna to Bits

WLAN/LTE

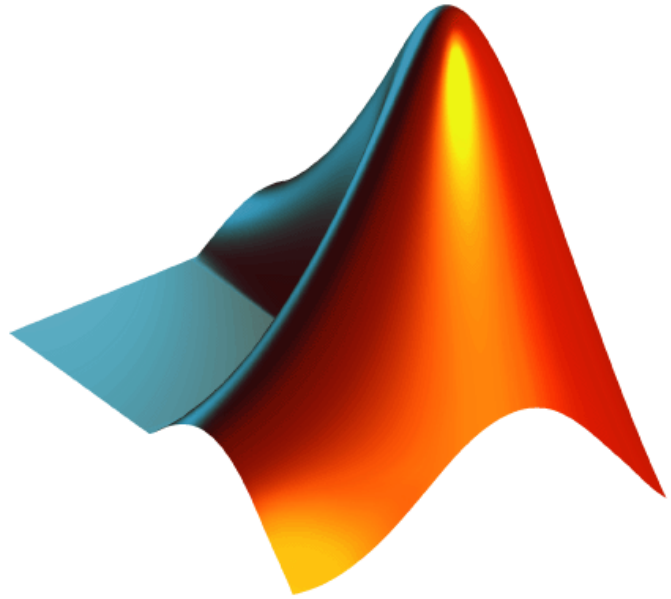
Image and Video Processing

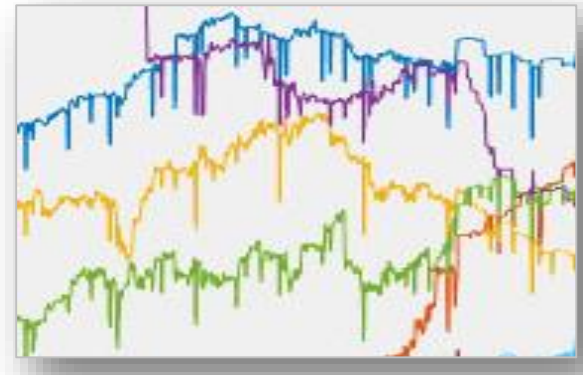
Signal Processing Engineers…

Signal Processing Engineers…



Develop algorithms

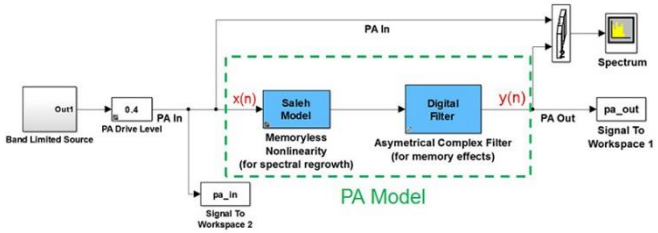




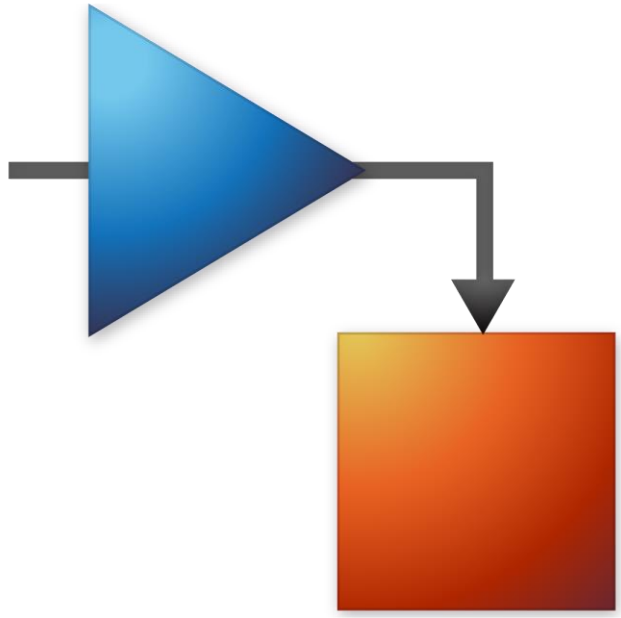
Analyze data

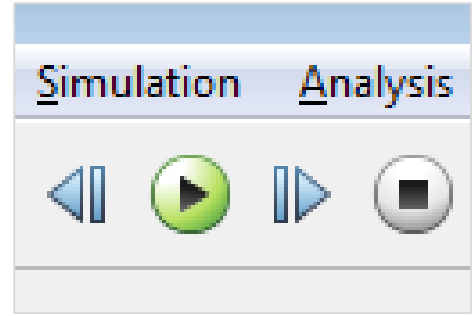
write MATLAB code.

Signal Processing Engineers…



Model systems

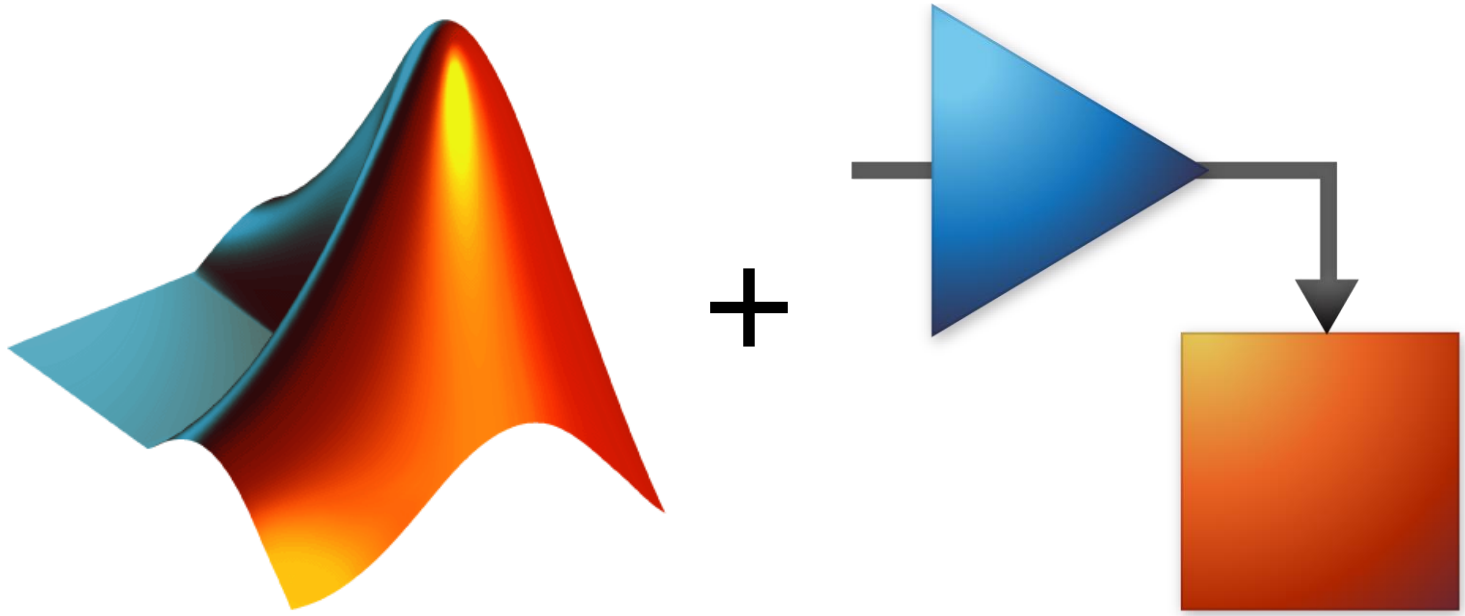




Run simulations

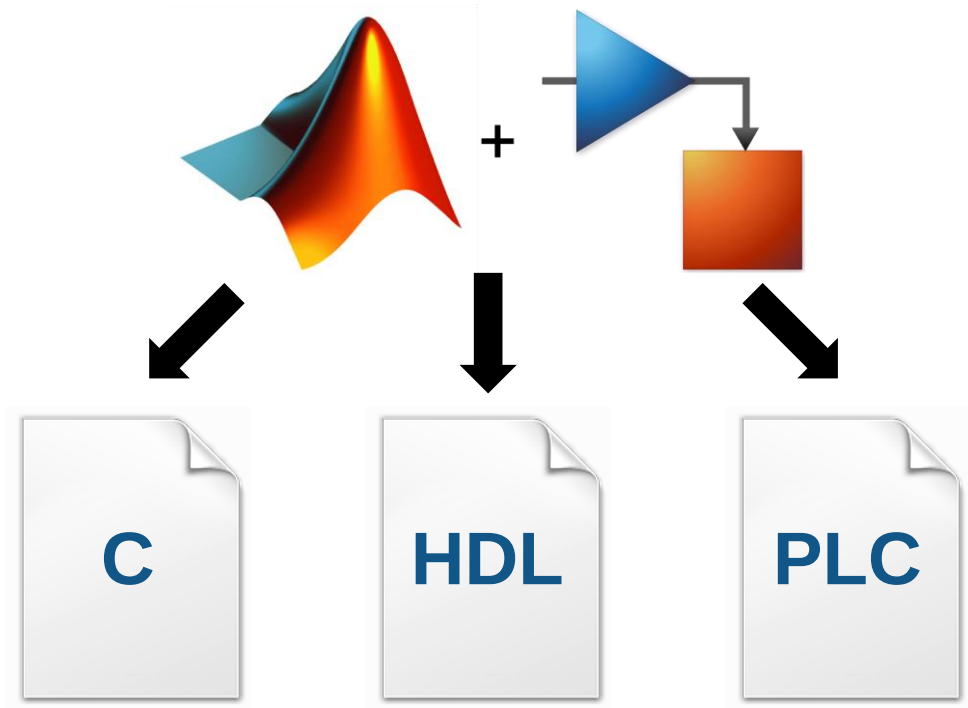
build Simulink models.

Signal Processing Engineers…



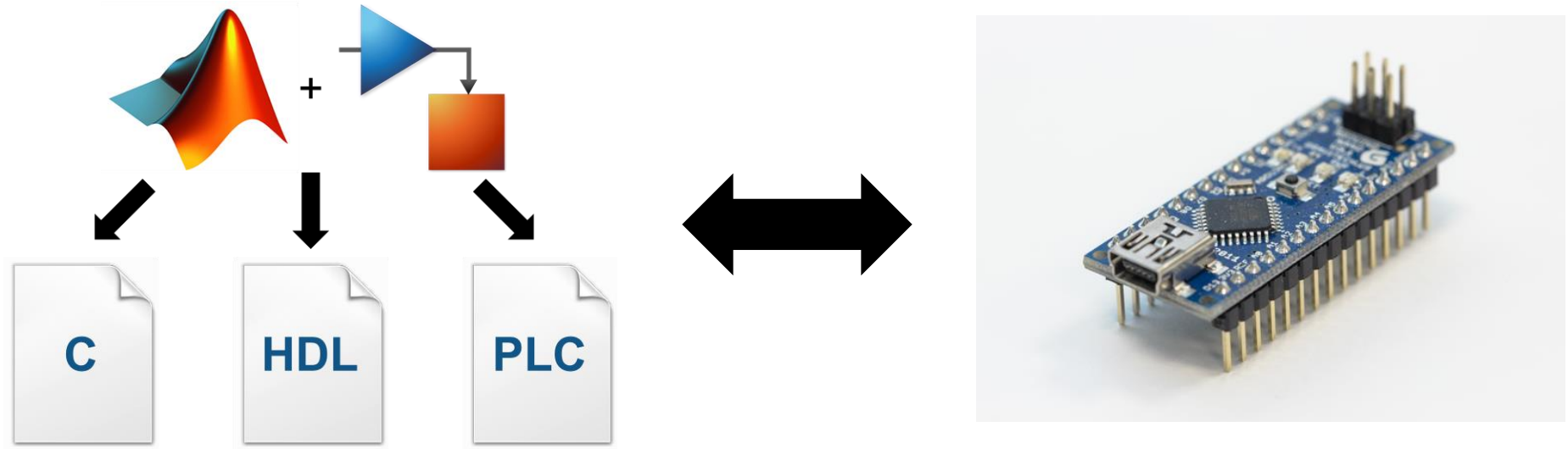
combine MATLAB code and Simulink models together.

Signal Processing Engineers…



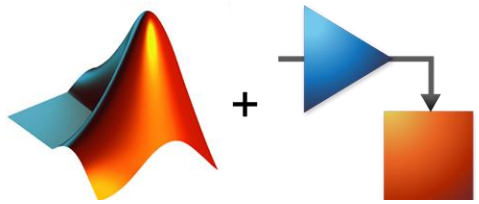
generate code.

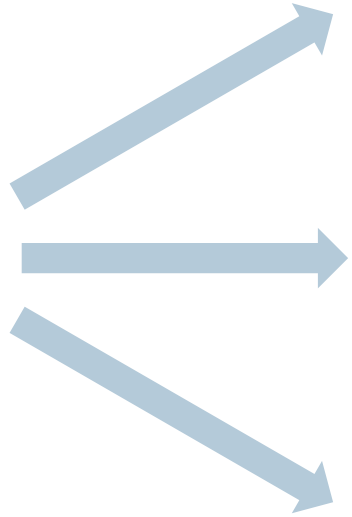
Signal Processing Engineers…

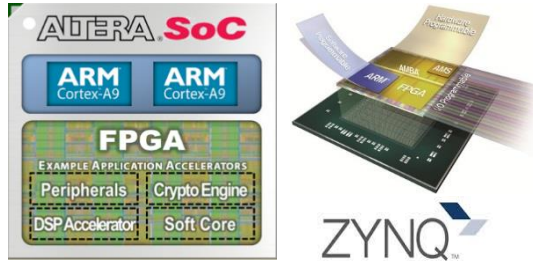


connect software to hardware.

General trend... | *Idea to implementation*



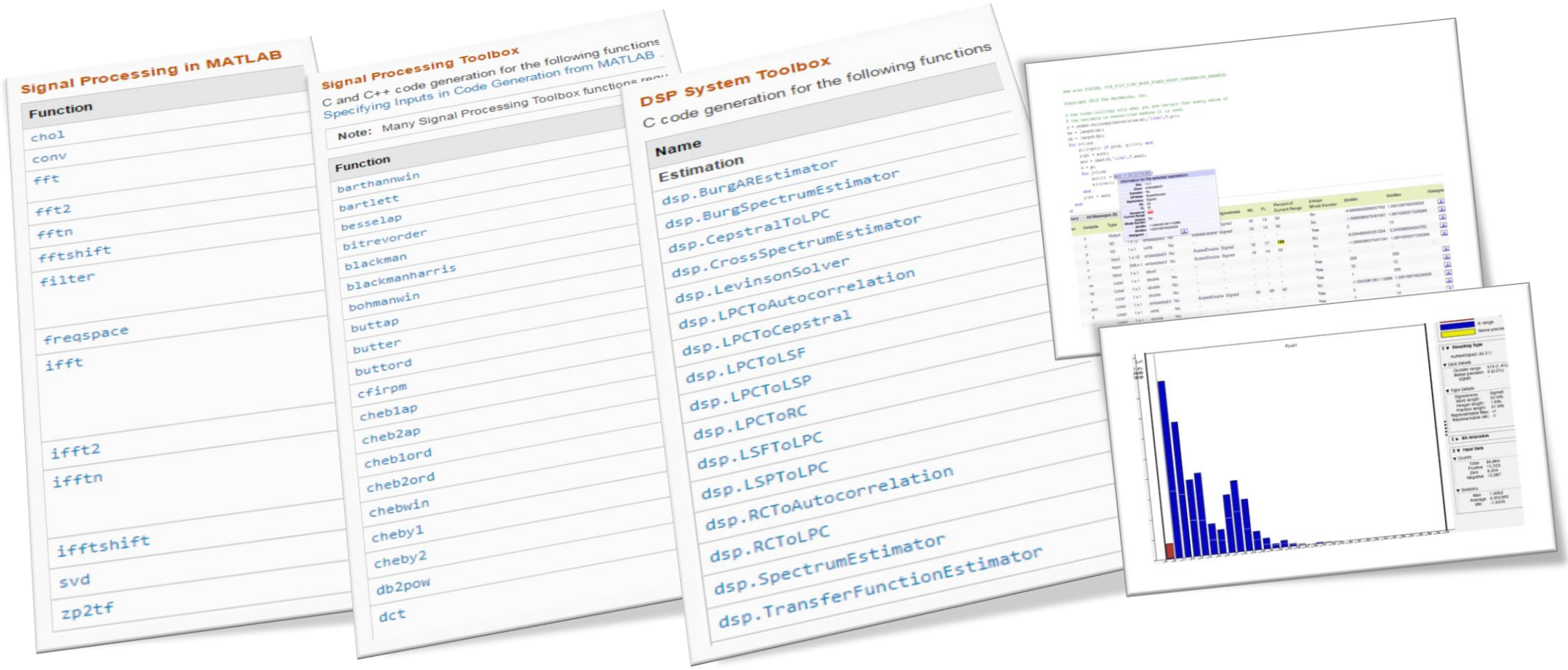




Increased support for code generation and fixed point design

Functions and Objects Supported for C and C++ Code Generation — Category List

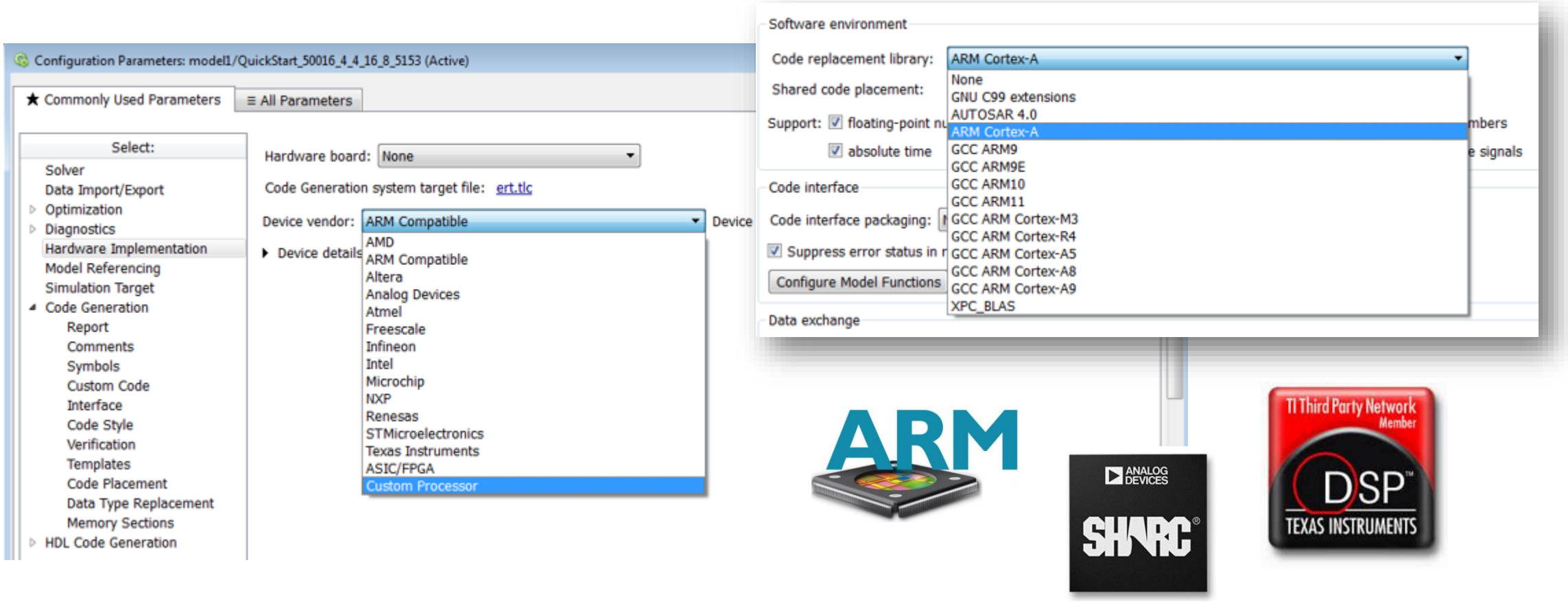
You can generate efficient C and C++ code for a subset of MATLAB® built-in functions and toolbox functions, classes, and System objects that you call from MATLAB code. These functions, classes, and System objects are listed by MATLAB category or toolbox category in the following tables.



Optimized libraries for DSPs

ARM Cortex-M and ARM Cortex-A Optimization R2016a

The DSP System Toolbox™ supports optimized C code generation for popular algorithms like FIR filtering and FFT on ARM® Cortex®-M and ARM Cortex-A processors.

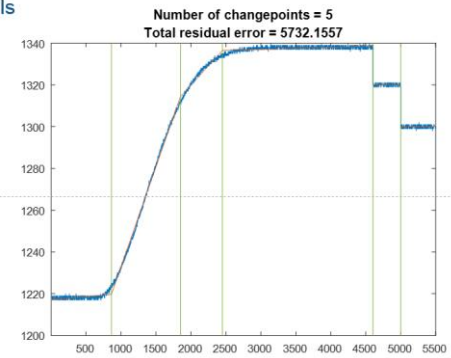


Some interesting additions...

Changepoint Detection

Find abrupt changes and statistical shifts in signals

- Determine “interesting” areas of an input signal
- Statistics supported
  - Mean
  - Variance
  - Mean and variance
  - Linear Regression



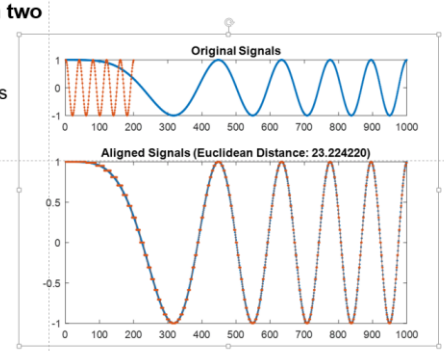

```
>>  load('engineRPM.mat','x')
>>  findchangepts(x,'Statistic','linear','MinThreshold',var(x))
```

Dynamic Time Warping

Stretch, align and compare signals with different time scales

Compare and align trajectories between two signals in space

- Obtain a measure of similarity of two signals trajectories.
- Optional time alignment
- Popular distance metrics supported:
  - Euclidean
  - Squared Euclidean
  - Manhattan
  - Symmetric Kullback-Leibler

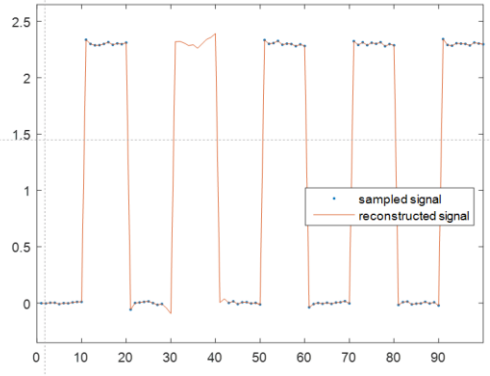



```
>> x = chirp(0:999,0,1000,1/100);
>> y = cos(2*pi*5*(0:199)/200);
>> dtw(x,y)
```

Gap Filling

Reconstruct missing samples using autoregressive modeling

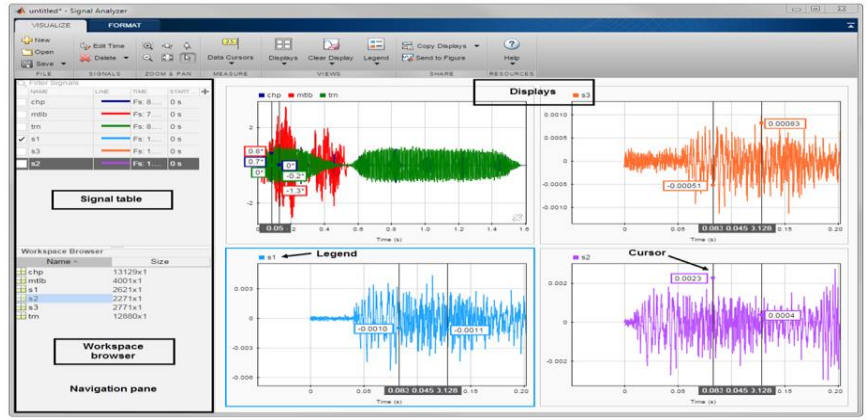
- Allows finer prediction for many input signals.
- Automatic model selection via Akaike information criterion
- Multiple gaps.
- Optionally model non-stationary signals




```
>> load clockex
>> x(29:42) = NaN;
>> fillgaps(x)
```

Signal Analyzer App

Visualize and compare multiple signals



Signal Processing

Audio

Antenna to Bits

WLAN/LTE

Image and Video Processing

Audio System Toolbox

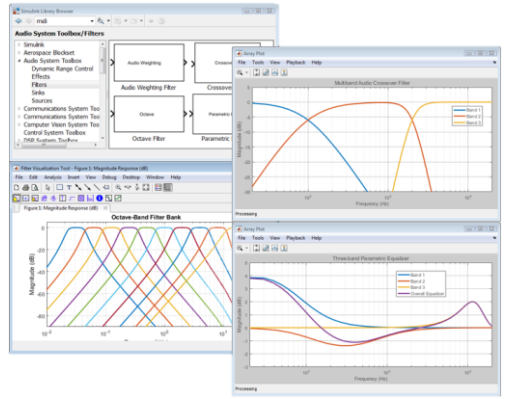
Design and test audio processing systems

MathWorks

R2016a

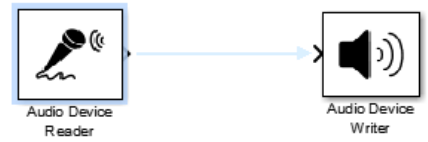
Audio System Toolbox

Design and test audio processing systems



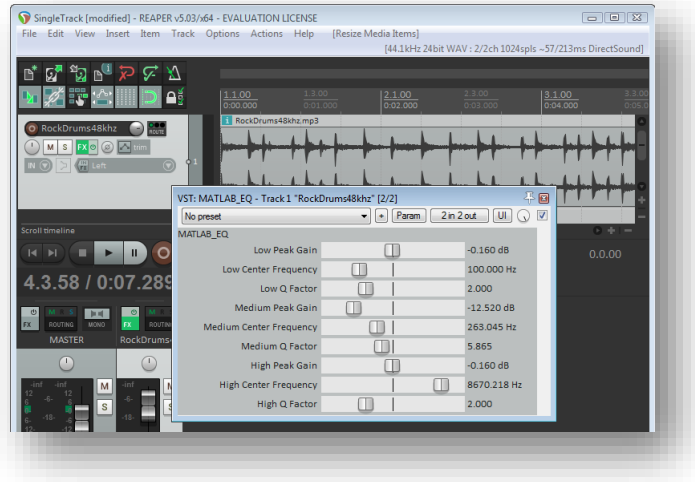
- Libraries of audio processing **algorithms** and examples

- **Low-latency audio streaming** from and to standard audio interfaces (e.g. ASIO, CoreAudio, ALSA)



- **Live-tuning** of MATLAB and Simulink via UI and MIDI controls

- **VST** plugin generation to run on Digital Audio Workstations



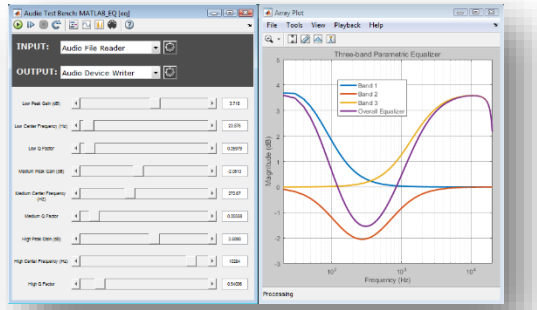
Audio System Toolbox

Prototyping for product development

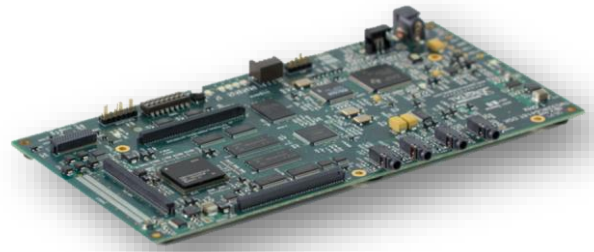
MATLAB algorithm

```
classdef VolumeControl < AudioPlugin    %#codegen
    properties
        % Public properties that are not constant and no
        % become tunable plugin parameters and appear on
        % dialog.  A default value is required.
        Volume = 1
    end
    properties (Constant)
        % Constant property 'DisplayName' specifies the
        % displayed by the DAW.
        DisplayName = 'Volume control'

        % Appearance and display of plugin parameters is
        % constant properties having the 'Display' suffi
        %     name displayed on the dialog
        %     units displayed on the dialog
        %     minimum value
        %     maximum value
        %     scaling law; one of {'lin' 'log' 'fader'
        %     scaling power (if law is 'pow')

        % Volume faders ordinarily vary the gain from so
        % silence (-inf dB, or 0V/V).  Here the maximum
        % The 'fader' law is cubic, which closely mimics
        % controls.
        VolumeDisplay = AudioPlugin.paramDisplay('Volume
    end
    methods
        function out = process(plug, in)
            % Processing to be applied to each frame of
            % Plugin input and output parameters are alw
            % channel; use multiple parameters to get mu
            % Frame size is not fixed, and may vary fro
            % Thus arguments will always be var-sized co
            % This method runs in the high priority audi
            % thread, so be as efficient as possible.
```

Early validation (listening tests)

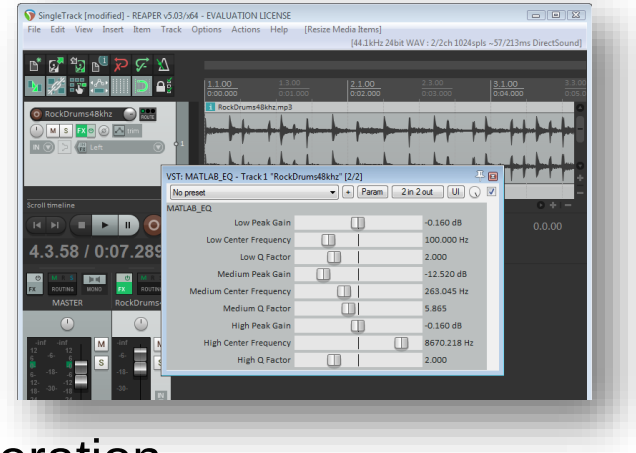


Advanced prototyping or production



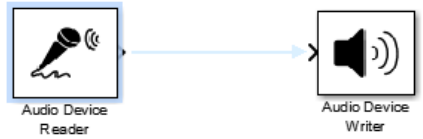
Audio System Toolbox

Use cases summary

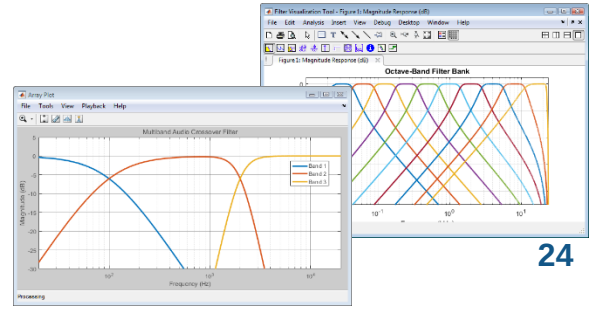


- **Desktop prototyping and listening tests**
  - Pain: prototyping costly and time-consuming
  - Solution: real-time audio streaming in MATLAB *and* VST plugin generation

- **Real-time custom measurements and signal analysis**
  - Pain: test & measurement equipment not available or not customizable
  - Solution: real-time audio acquisition and *unlimited* custom analysis



- **Audio algorithm design**
  - Pain: re-inventing consolidated algorithms time-consuming
  - Solution: libraries of audio processing algorithms and examples

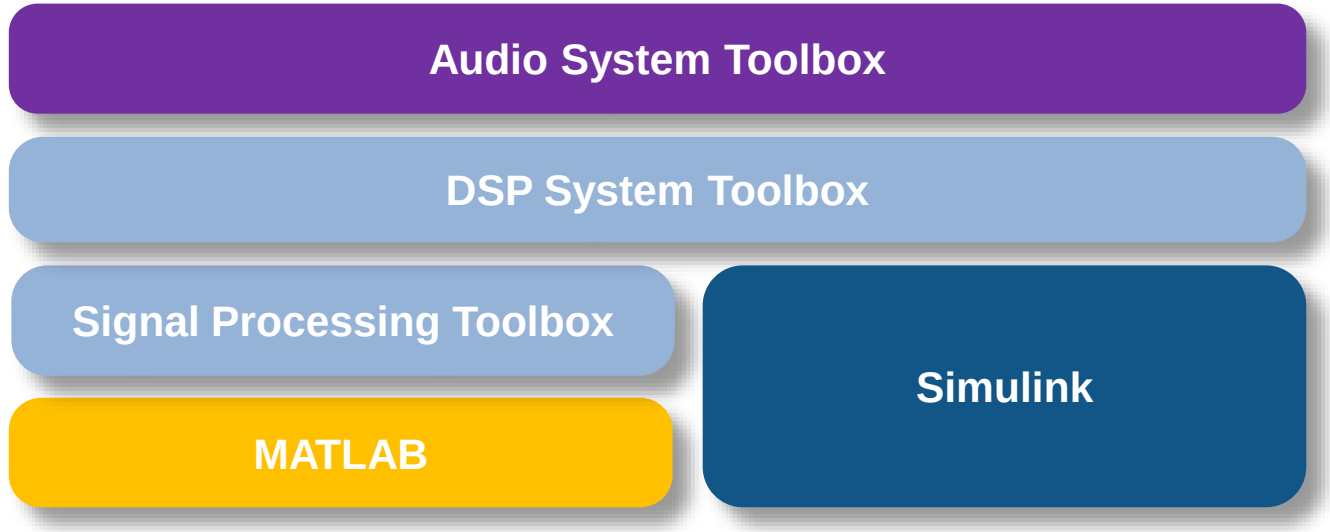


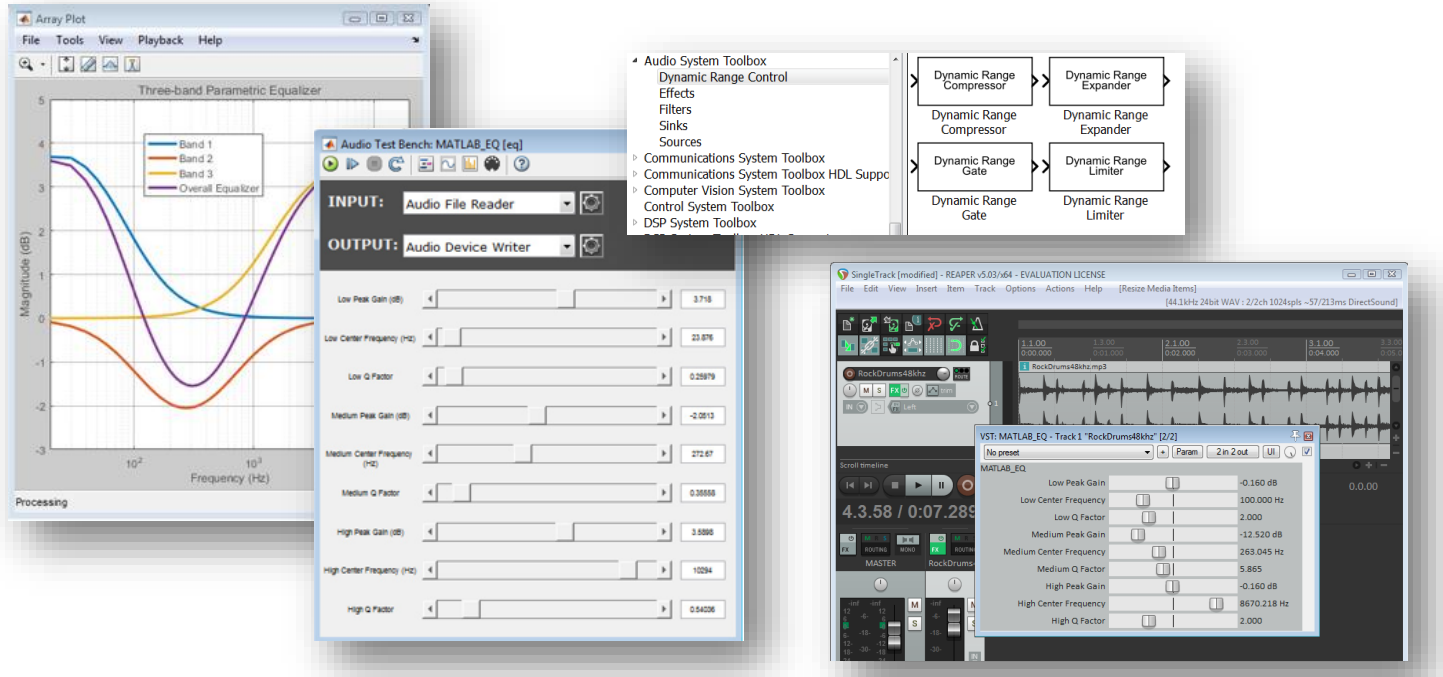
Audio System Toolbox

Product ecosystem

- **Requires**
  - MATLAB
  - Signal Processing Toolbox
  - DSP System Toolbox

- **Supports**
  - MATLAB
  - Simulink
  - C/C++ Code Generation





Signal Processing

Audio

Antenna to Bits

WLAN/LTE

Image and Video Processing

Antenna to Bits

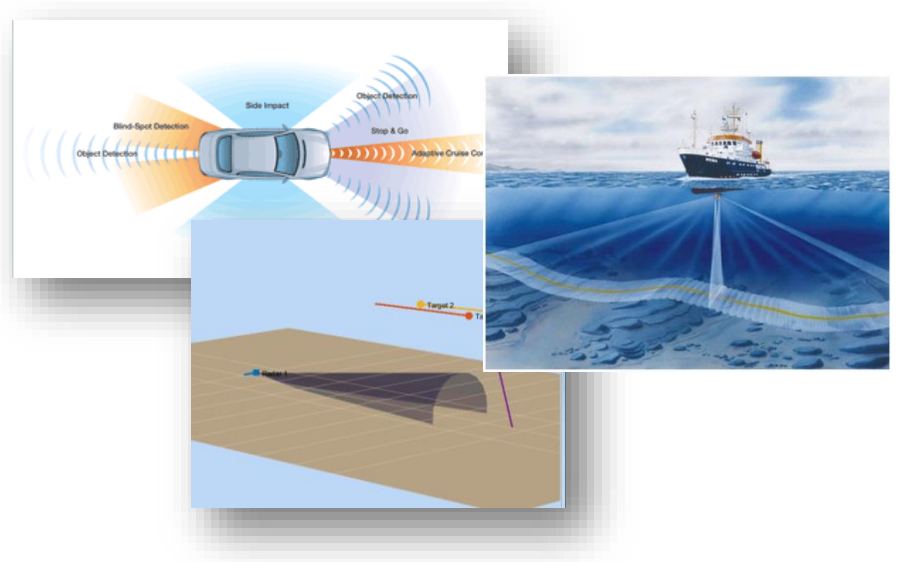
System Design and Modelling

Communications Systems

- System Partitioning
- Link Budget Simulations
- System Integration

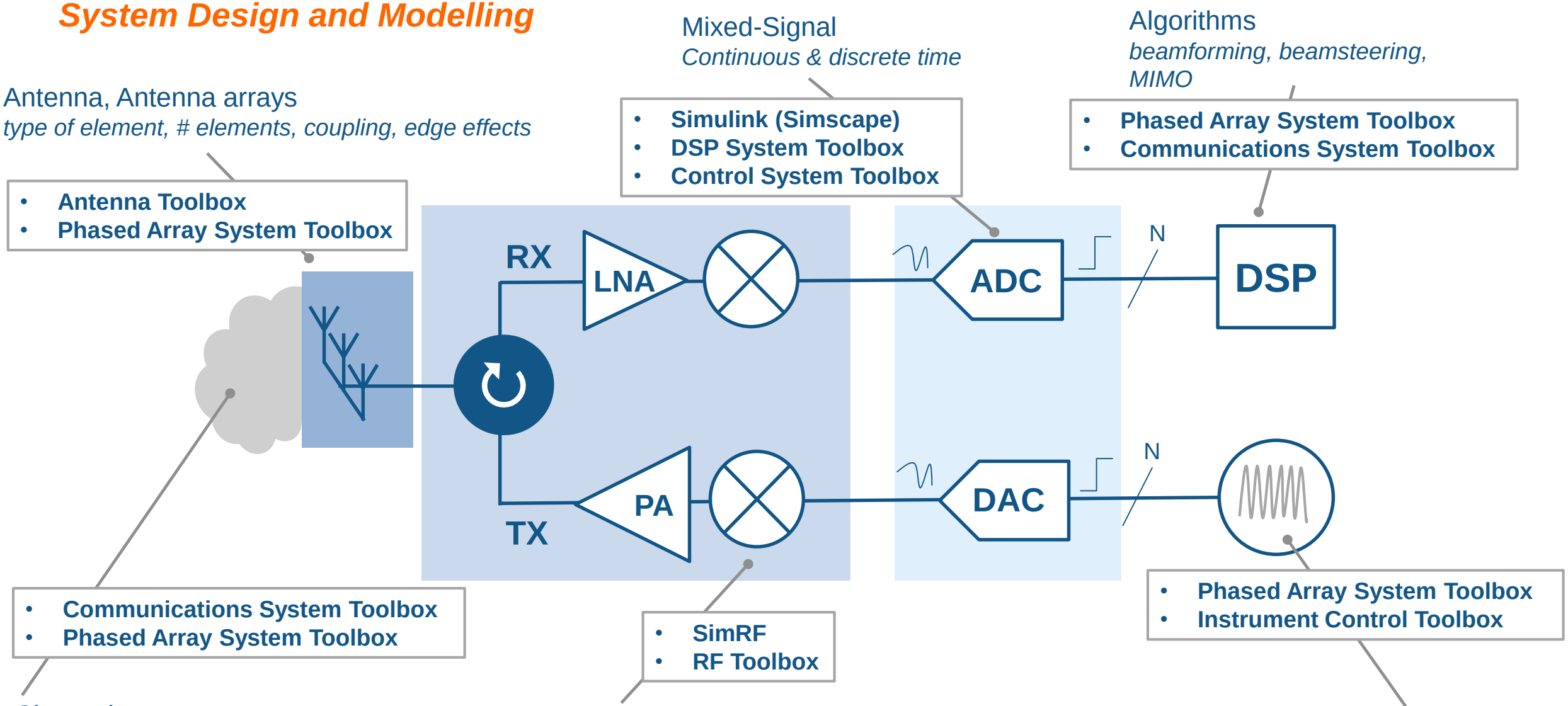
- Elaborating RF Architecture
- Component Simulation
- RF Subsystem Simulation

Radar / Sonar / Sensor Arrays



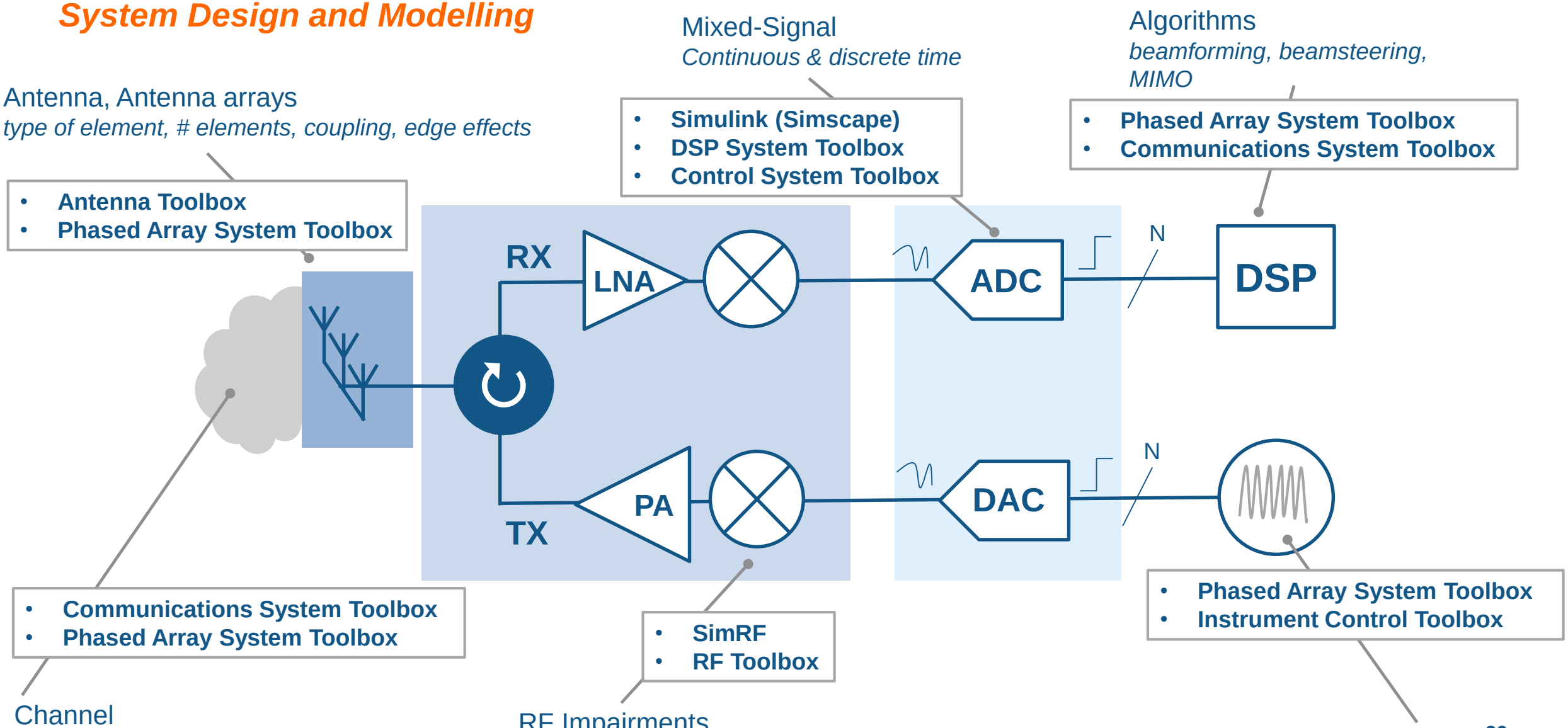
Antenna to Bits

System Design and Modelling



Antenna to Bits

System Design and Modelling



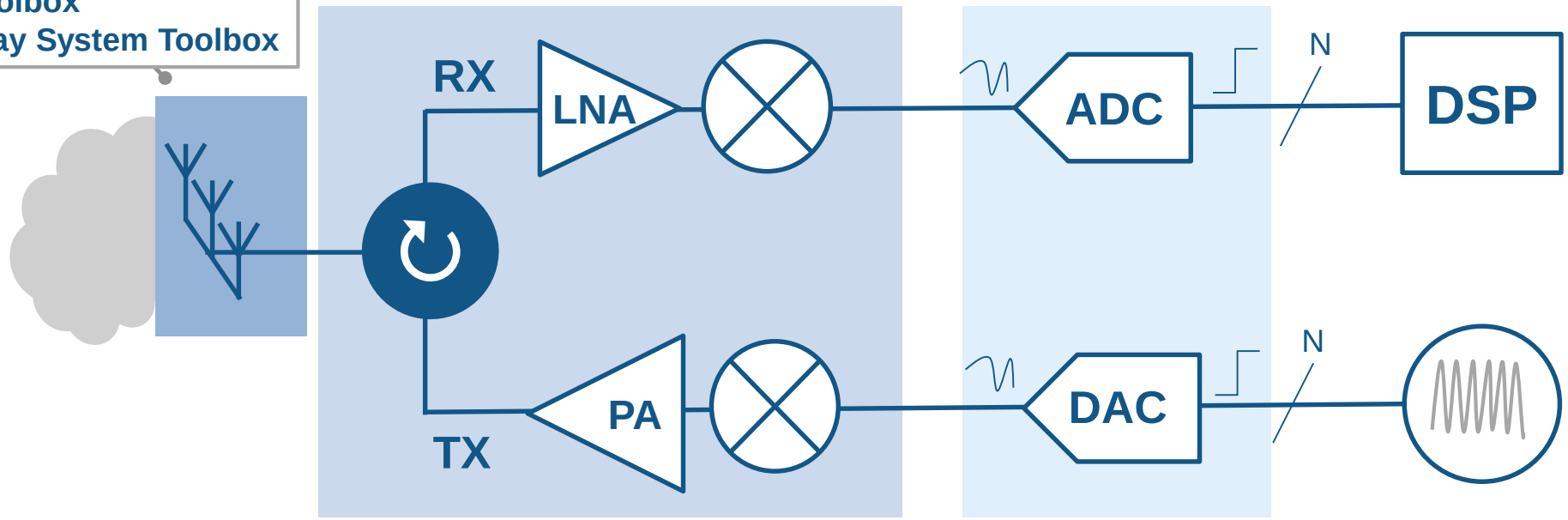
Antenna to Bits

System Design and Modelling

Antenna, Antenna arrays

type of element, # elements, coupling, edge effects

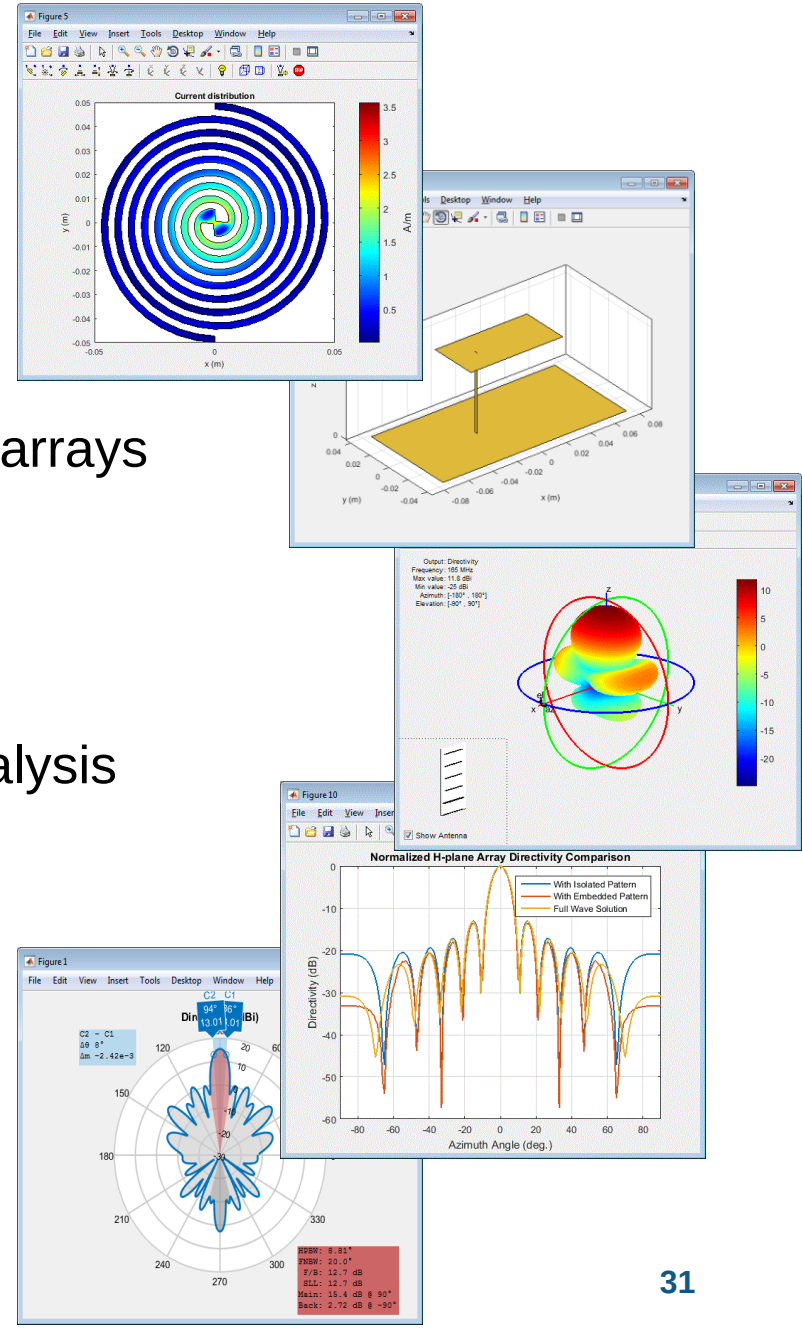
- **Antenna Toolbox**
- **Phased Array System Toolbox**



Antenna Toolbox

Design, simulation and integration

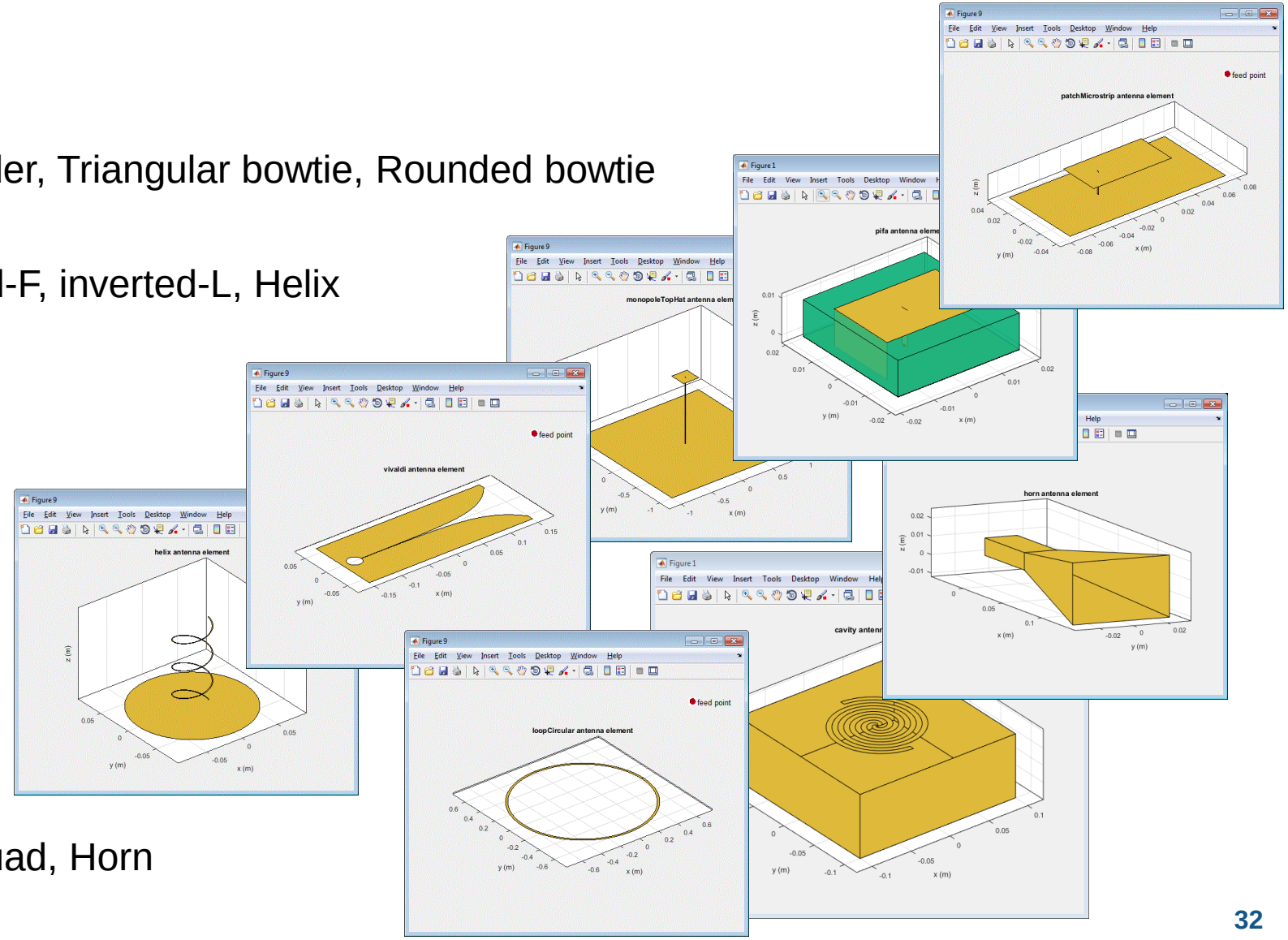
- Easy **design**
  - Library of parameterized antenna elements
  - Functionality for the design of linear and rectangular antenna arrays
  - No need for full CAD design
- Rapid **simulation** setup
  - Method of Moments field solver for port, field, and surface analysis
  - No need to be an EM expert
- Seamless **integration**
  - Model the antenna together with signal processing algorithms
  - Rapid iteration of different antenna scenarios for radar and communication systems design



Antenna Toolbox

Library of Available Geometries

- Dipole antennas
  - Dipole, Vee, Folded, Meander, Triangular bowtie, Rounded bowtie
- Monopole antennas
  - Monopole, Top hat, Inverted-F, inverted-L, Helix
- Patch antennas
  - Microstrip patch, PIFA
- Spirals
  - Equiangular, Archimedean
- Loops
  - Circular, rectangular
- Backing structures
  - Reflector and cavity
- Other common antennas
  - Yagi Uda, Slot, Vivaldi, Biquad, Horn



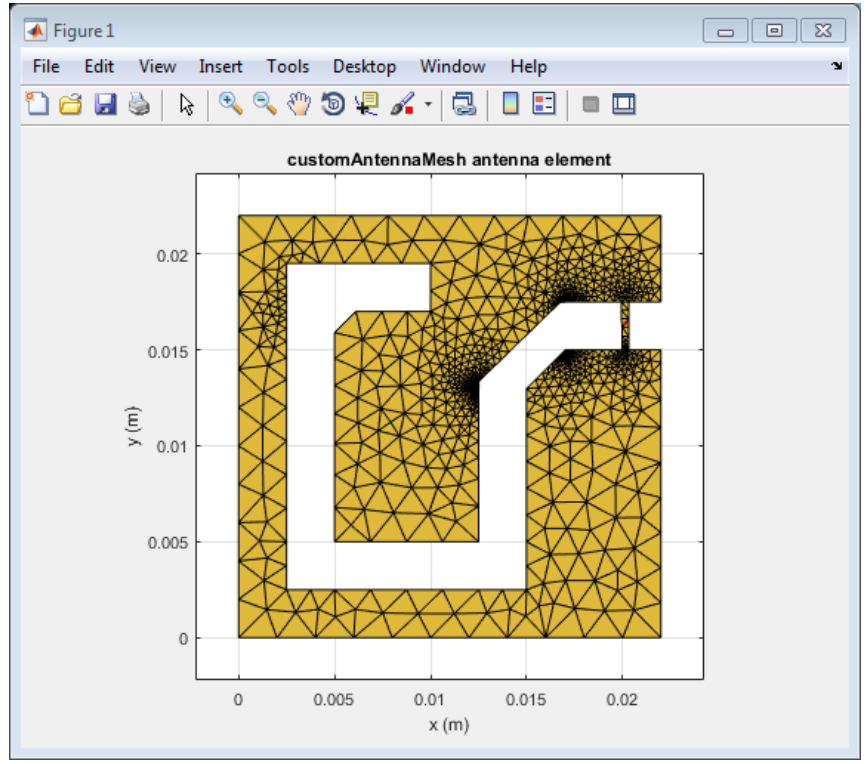
Antenna Toolbox

Custom Antenna Element Design

- Define your custom planar structure
  - Define the antenna geometry using PDE Toolbox
  - Define the mesh using MATLAB

    `delaunayTriangulation`
  - Use third party tools to generate a mesh structure

- Import 2D mesh with Antenna Toolbox
  - Define the feeding point
  - Analyse the antenna



Antenna Toolbox

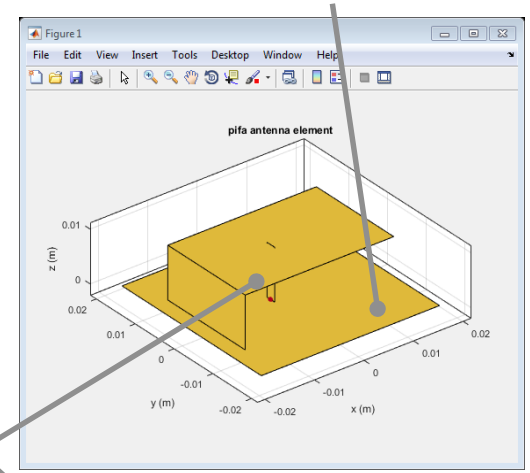
Dielectric Substrate Modelling

- Antenna are often mounted on **substrates**
- Dielectric properties:

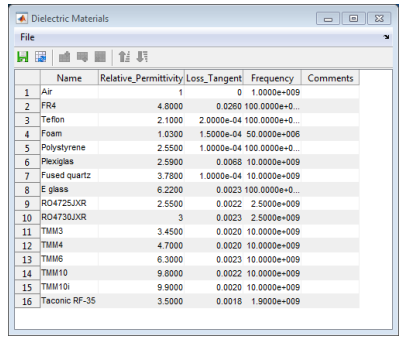
Dielectric	Relative permittivity	Loss Tangent
Air	1	0
Other	>1 (typically <10)	>0 (typically ~1e-3)

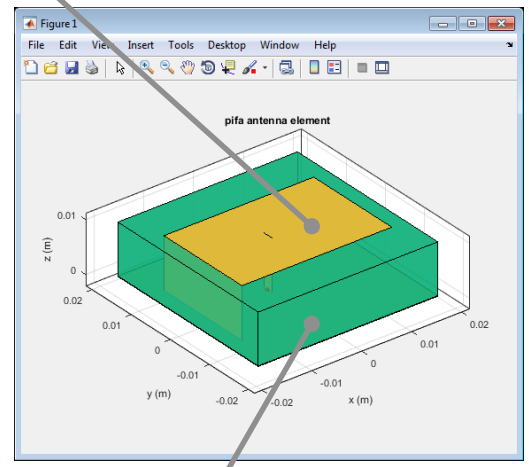
- Dielectric properties **affect resonance, bandwidth, efficiency, pattern** …
- Use the dielectric catalogue listing existing materials
- Define your **own** dielectric material

Free space (isolation)



“metal” antenna (ideal conductor)





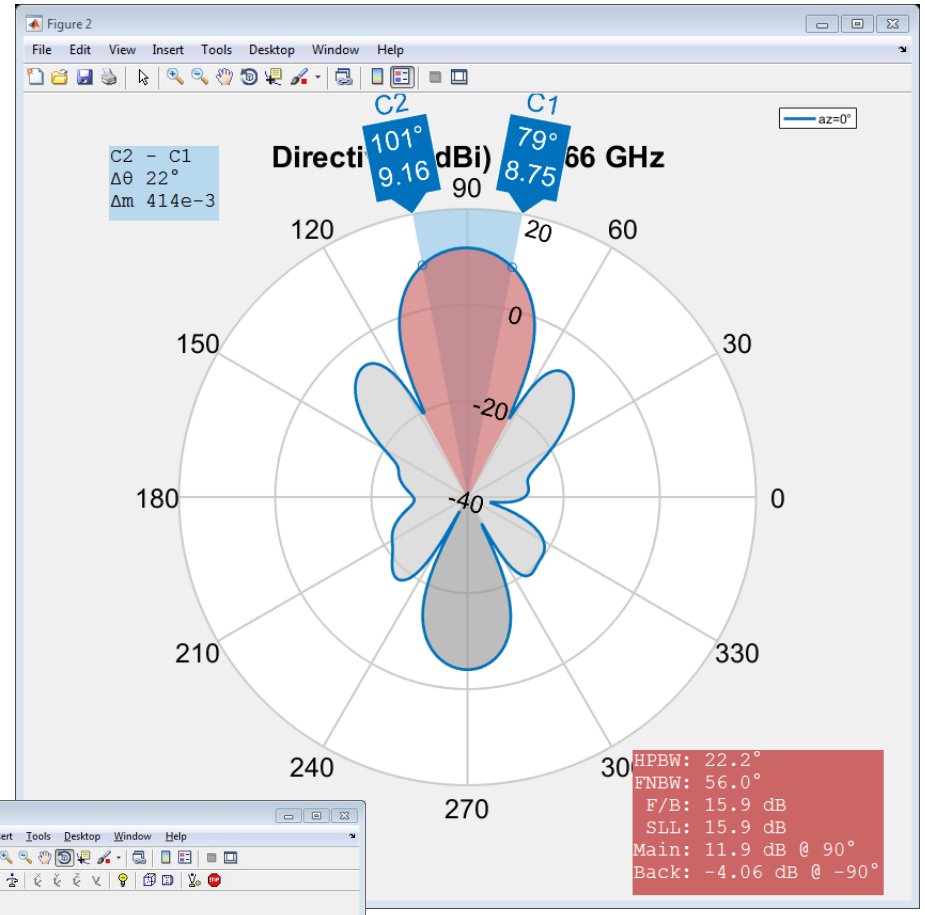
Dielectric substrate

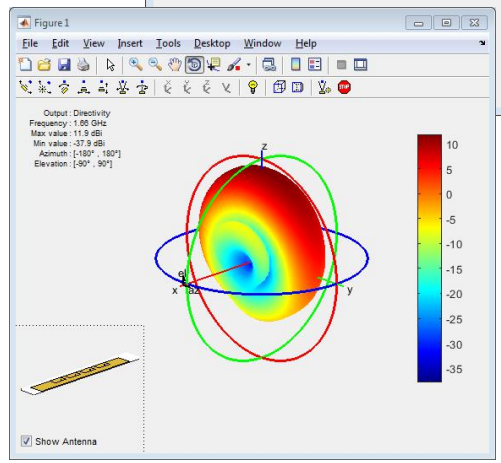
From antenna element to antenna array…

Phased Array System Toolbox

Array Antenna Design

```
>> a = linearArray

>> a.Element = p;

>> a.ElementSpacing = 0.1;

>> a.NumElements = 4;

>> layout(a);

>> patternElevation(a, 1.66e9,0);
```



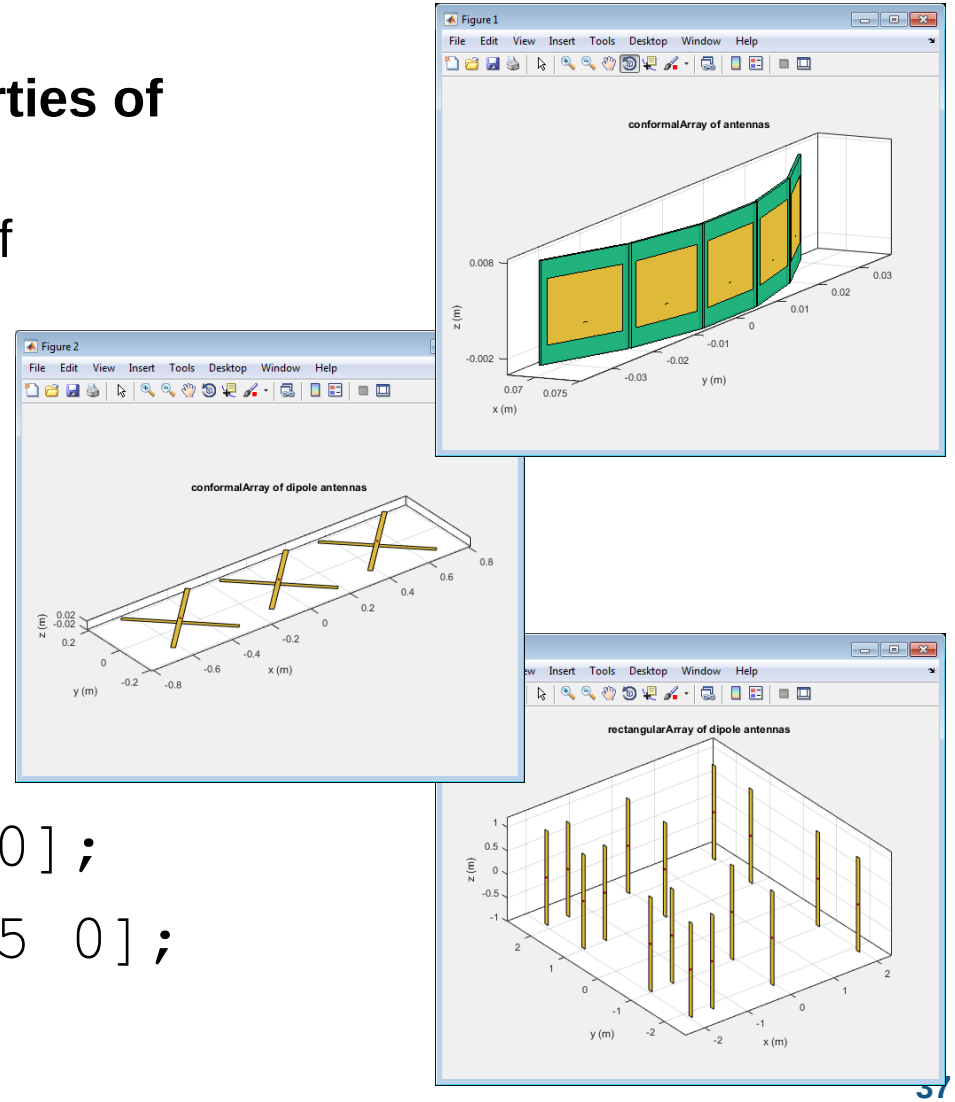


Phased Array System Toolbox

Custom Array Antenna Design

- Build regular arrays where you can change the **properties of individual elements** (rotation, size, tapering)
- Describe conformal (heterogeneous) arrays in terms of element type and arbitrary position

```
>> arr = conformalArray;
>> d = dipole;
>> b = bowtieTriangular;
>> arr.Element = {d, b};
>> arr.ElementPosition(1,:) = [0 0 0];
>> arr.ElementPosition(2,:) = [0 0.5 0];
```

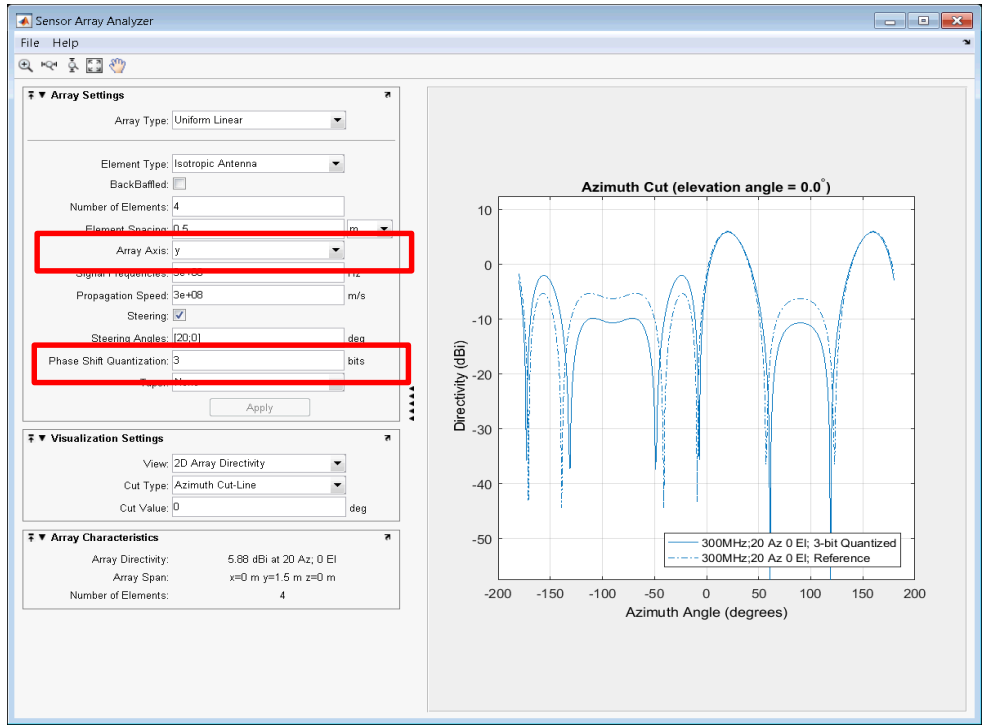


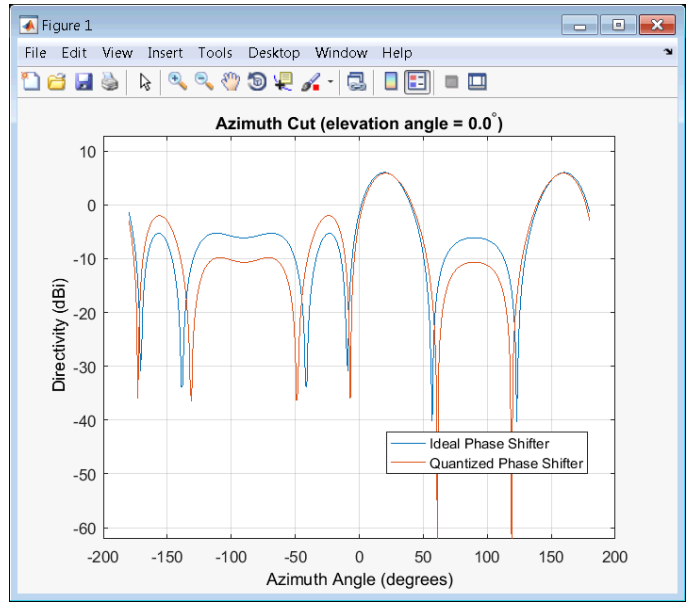
Phased Array System Toolbox

Model effects of quantized phase shift values on array patterns and responses

- Many phase shifters in real systems are quantized
- Allow customer to quickly see the effect of phase shifter quantization

```
ant = phased.ULA(4);
sv = phased.SteeringVector('SensorArray',ant);
w1 = step(sv,3e8,[20;10]);
release(sv);
sv.NumPhaseShifterBits = 3;
w2 = step(sv,3e8,[20;10]);
c = sv.PropagationSpeed;
pattern(ant,3e8,-180:180,0,'PropagationSpeed',c,'Weights',[w1 w2],...
    'CoordinateSystem','rectangular');
legend('Ideal Phase Shifter',...
    'Quantized Phase Shifter','Location','Best')
```





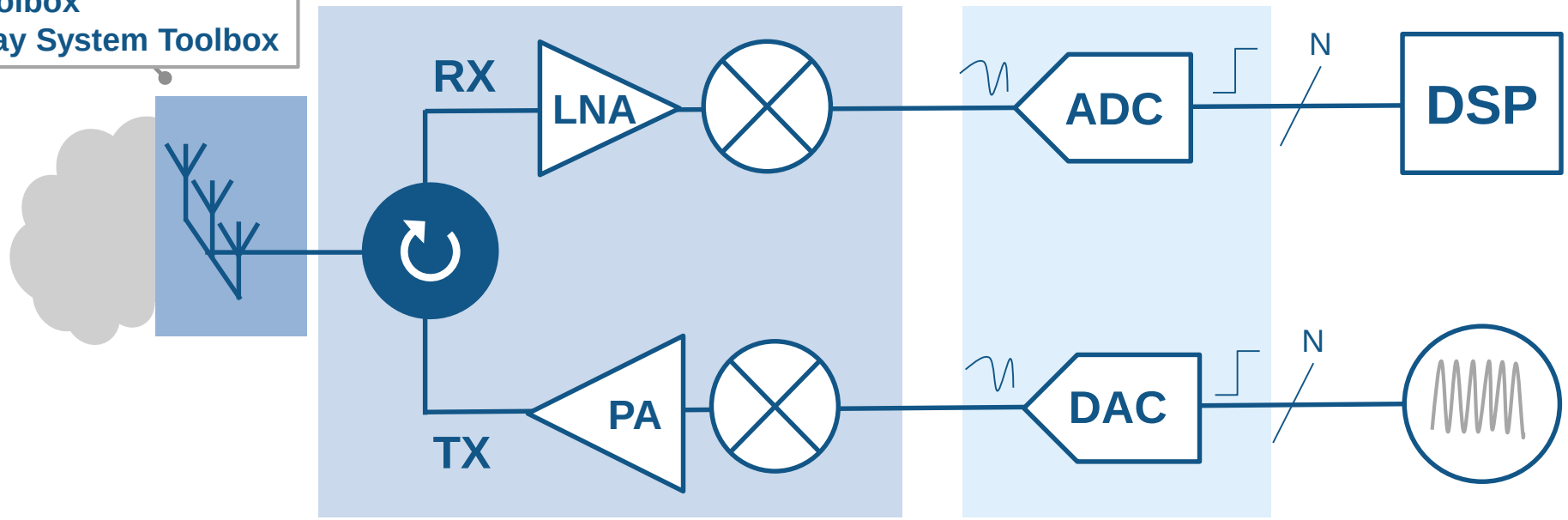
Antenna to Bits

System Design and Modelling

Antenna, Antenna arrays

type of element, # elements, coupling, edge effects

- **Antenna Toolbox**
- **Phased Array System Toolbox**



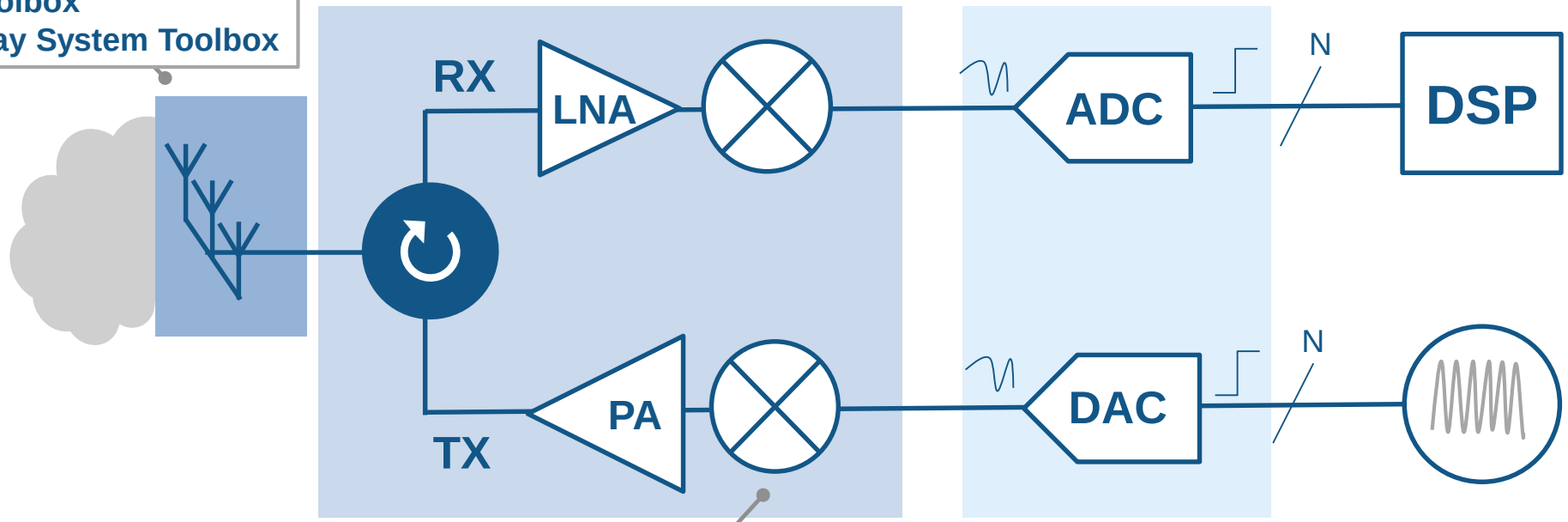
Antenna to Bits

System Design and Modelling

Antenna, Antenna arrays

type of element, # elements, coupling, edge effects

- **Antenna Toolbox**
- **Phased Array System Toolbox**



- **SimRF**
- **RF Toolbox**

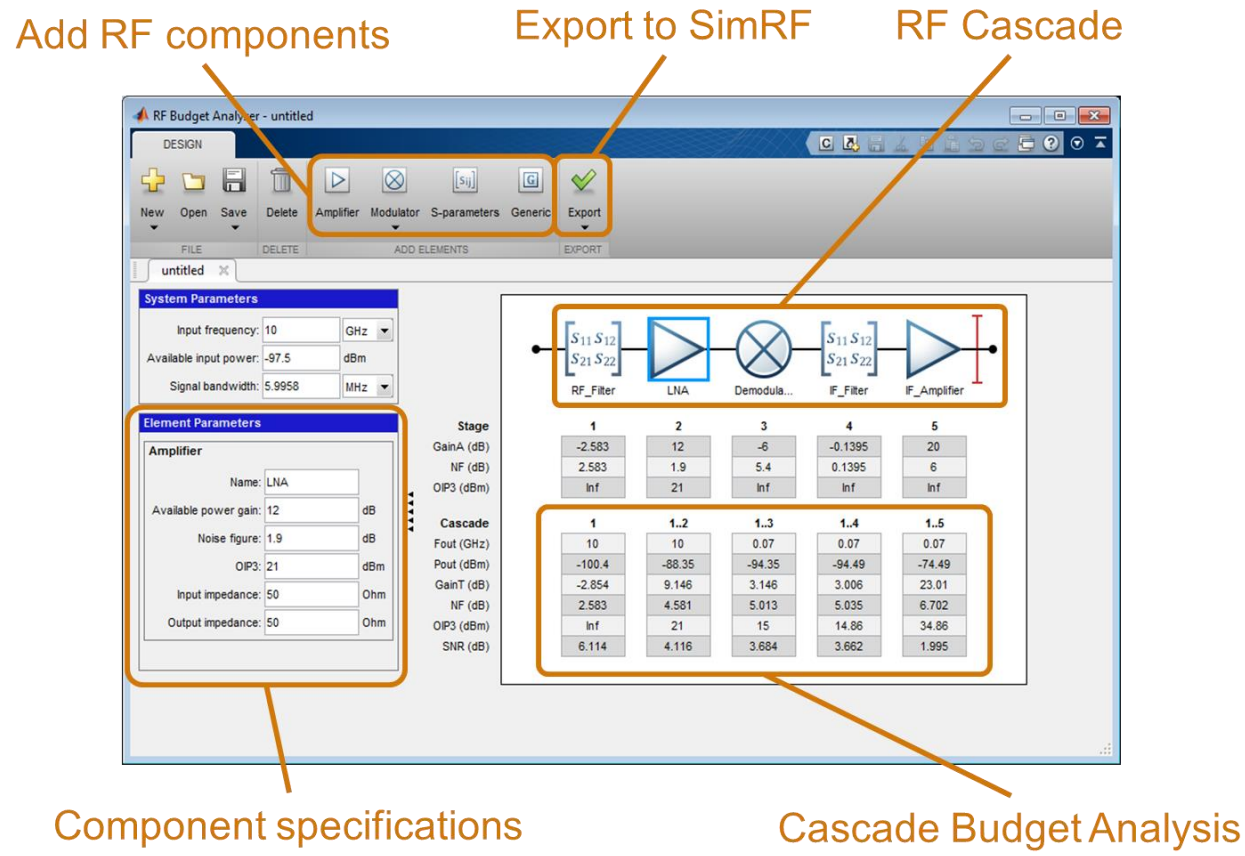
RF Impairments

frequency dependency, non-linearity, noise, mismatches

RF Toolbox

RF Budget Analyzer

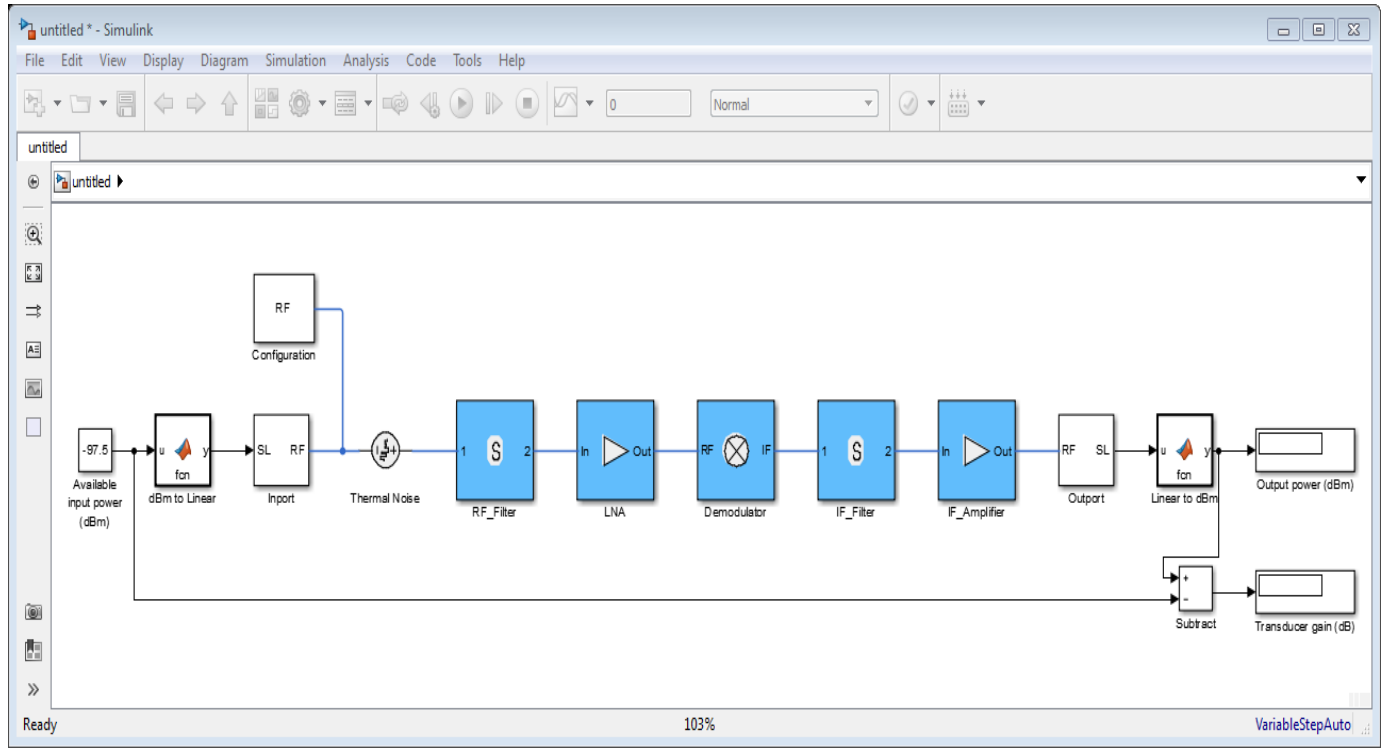
- Analytically compute gain, noise figure, and IP3 for cascaded RF components
- Specify components in terms of data sheet parameters and S-parameters
- Analyse the RF chain taking into account impedance mismatches



RF Toolbox

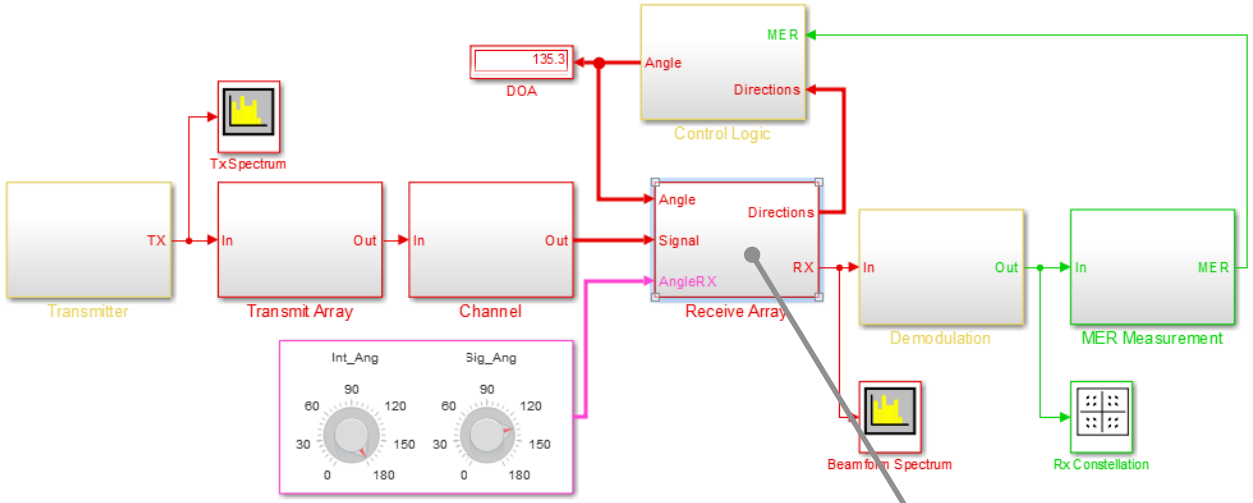
RF Budget Analyzer | Export to Sim RF

- Automatic testbench and SimRF model generation using the RF Budget Analyser App
- Validate simulation results using analytical computations
- Rapidly get started with Circuit Envelope simulation



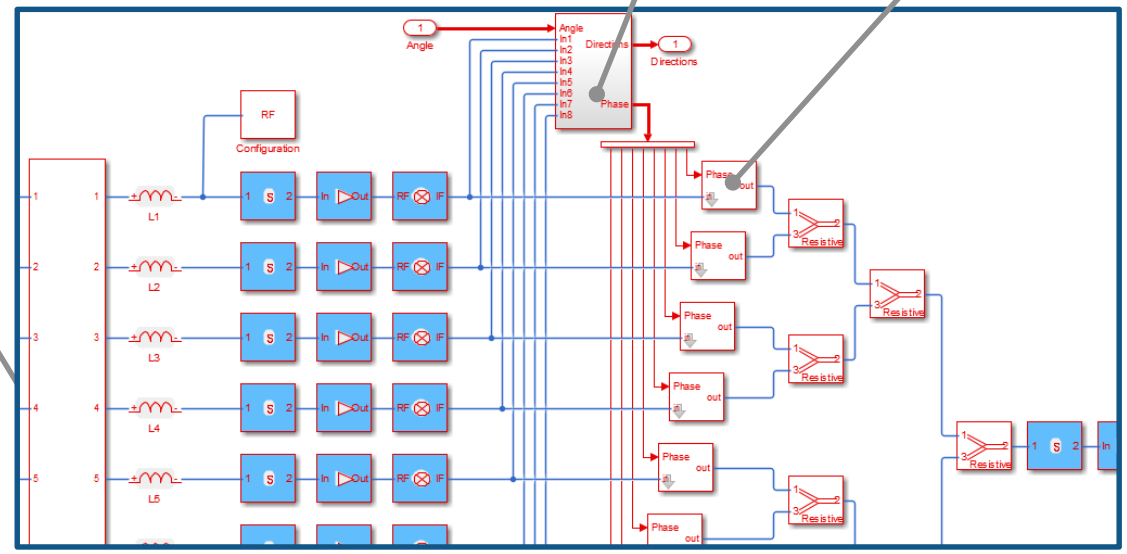
Sim RF

Example | MIMO Front End with RF Beamforming



Estimation of direction of arrival

RF phase shifting

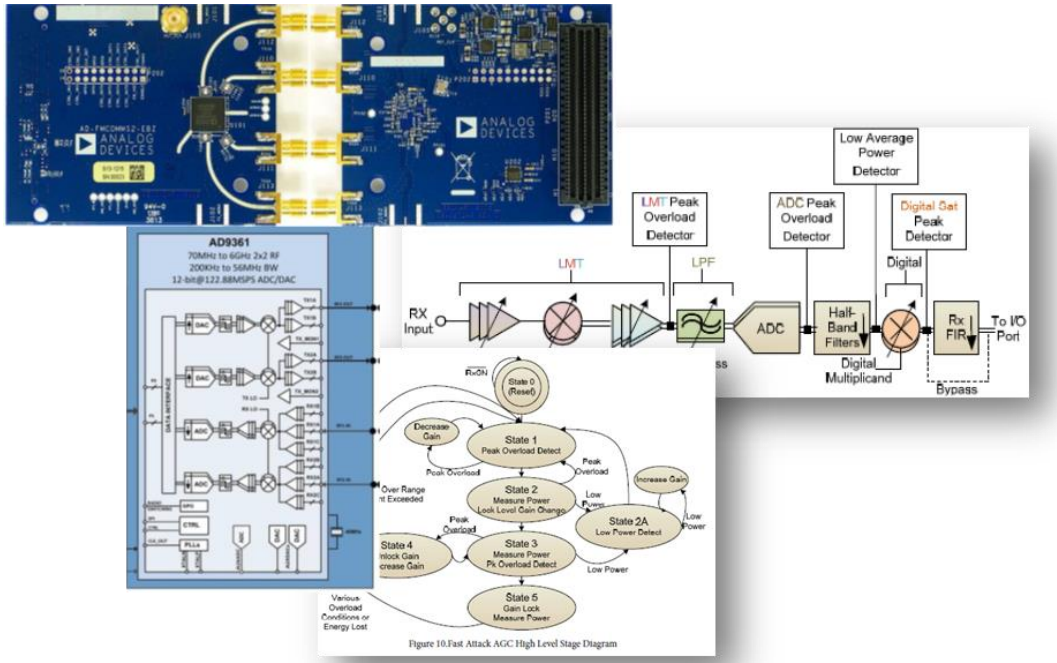


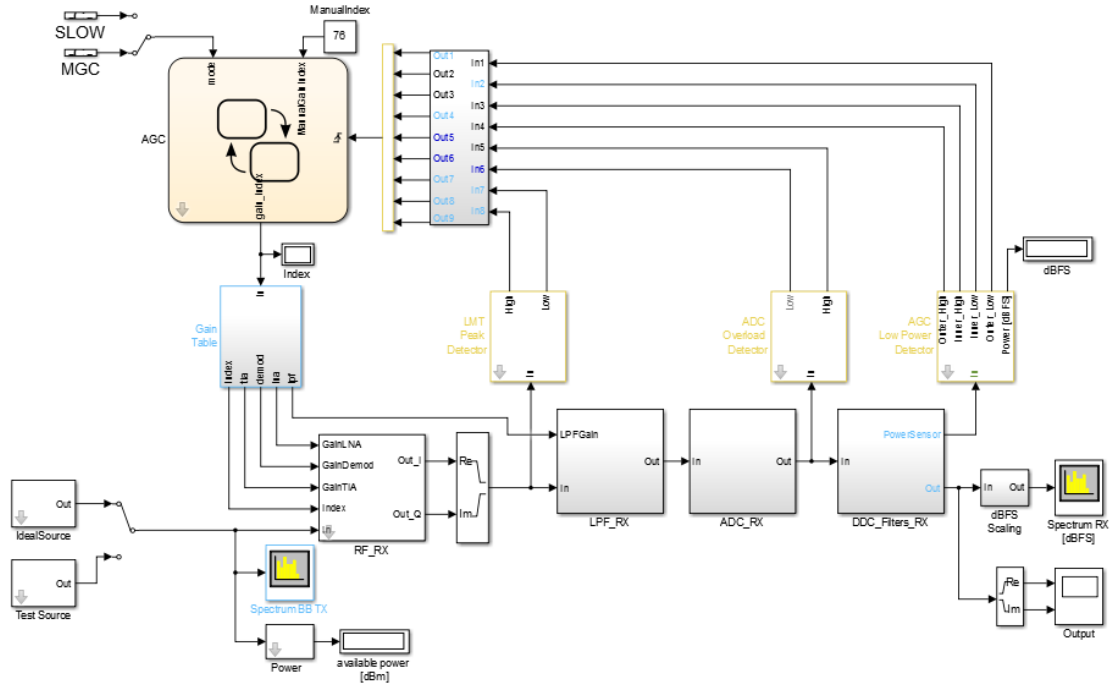
- Antenna coupling and loading (S-parameters)
- Antenna matching network
- RF and IF Filters described with Touchstone files
- IF demodulation with image rejection
- Non-linearity of the amplifiers
- Thermal Noise
- RF phase shifting and signal combiners

Sim RF

New and faster implementation of the AD9361 transmitter and receiver

AD9361
RF Agile Transceiver™
70 MHz – 6000 MHz Turning range
200kHz – 56 MHz RF channel Bandwidth

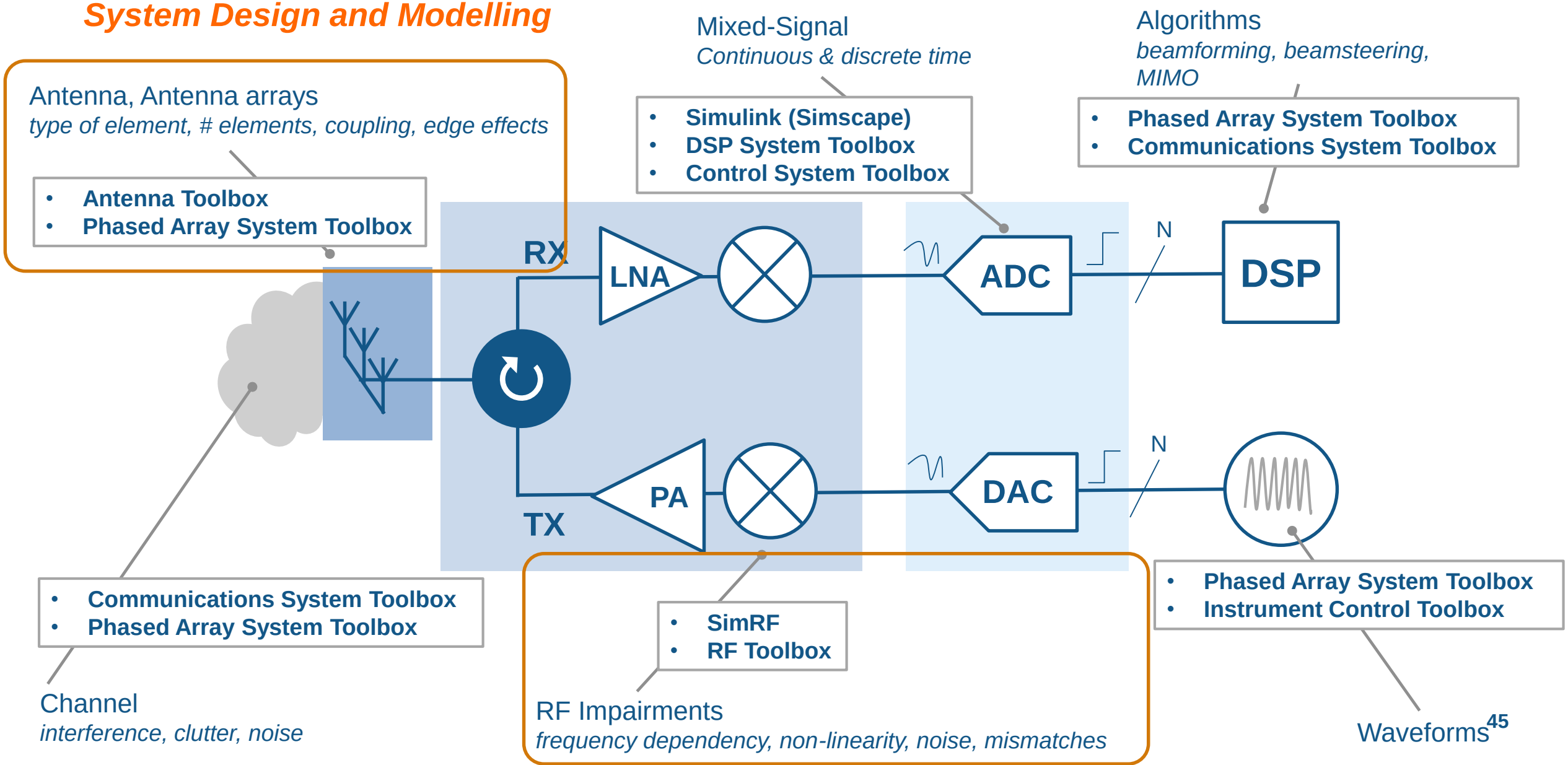




http://www.mathworks.com/adi-rf

Antenna to Bits

System Design and Modelling



Signal Processing

Audio

Antenna to Bits

WLAN/LTE

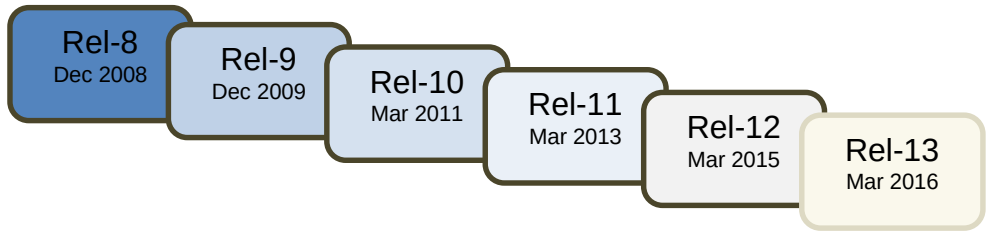
Image and Video Processing

WLAN/LTE and beyond…

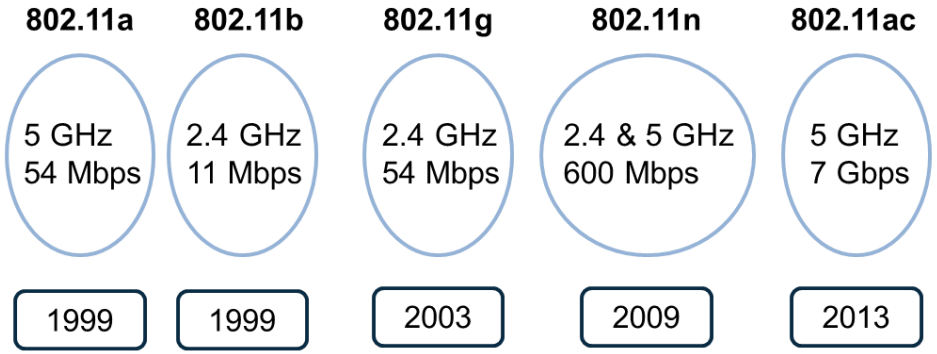
Evolution of Air Interface Technologies

4G

3GPP LTE, LTE-A



IEEE 802.11 WLAN standards



5G?

5G standardization

Requirements

- Higher data rates
- More flexible spectrum use
- Spatial resource
- Low delay & link adaptability
- Reliable service everywhere

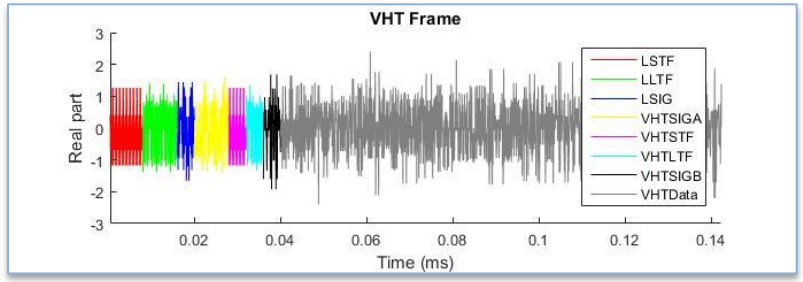
Proposed enabling technologies

- Massive MIMO
- Small Cell, HetNet
- New Modulations
- New Frequency bands

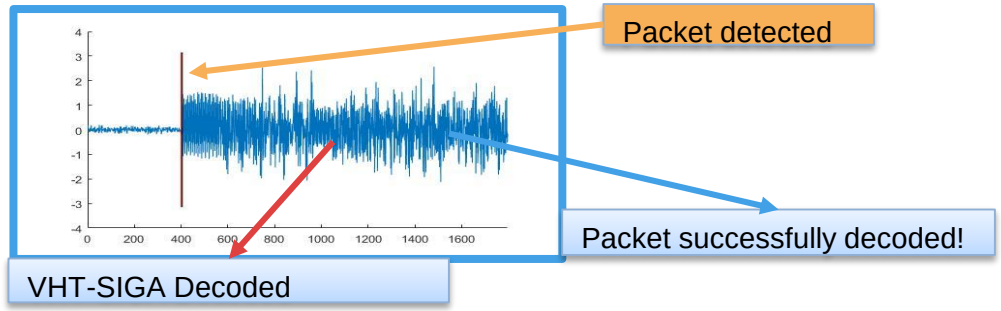
WLAN/LTE

Workflow/Use-cases of wireless designers

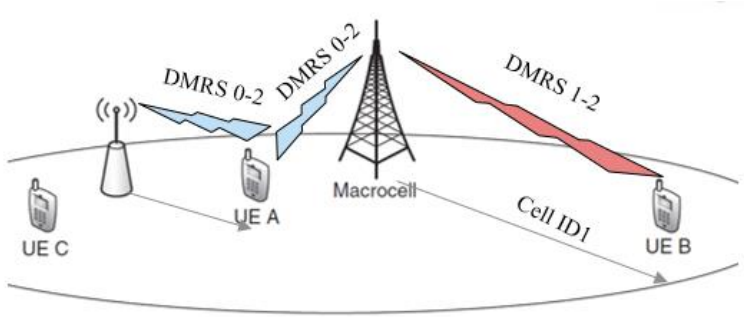
Signal Generation



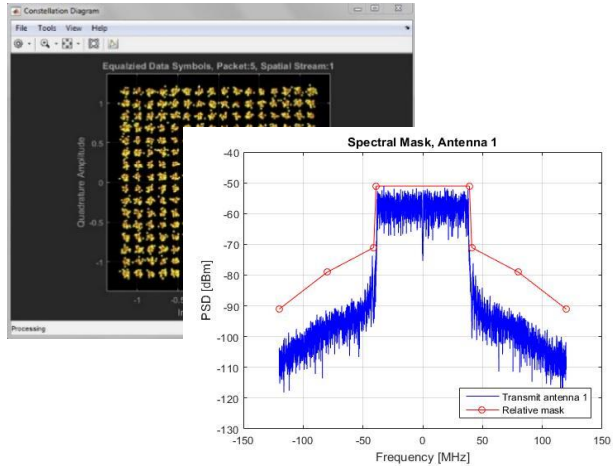
Signal Detection



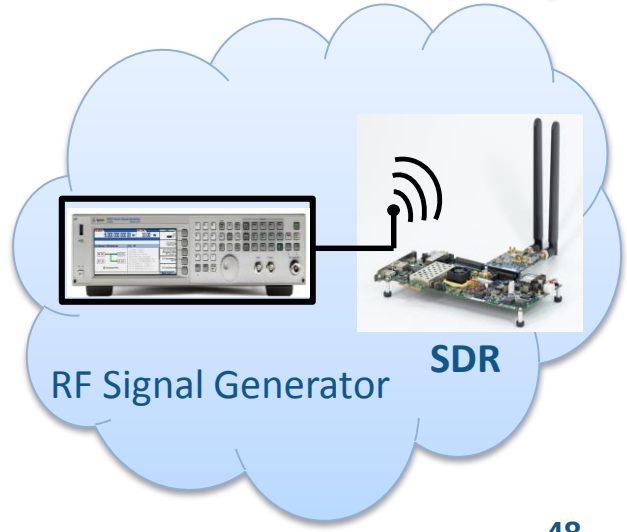
End-to-End Simulations



Measurements

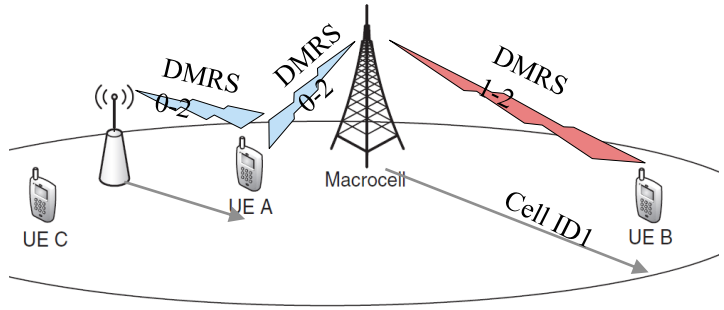


HW & Radio Connectivity

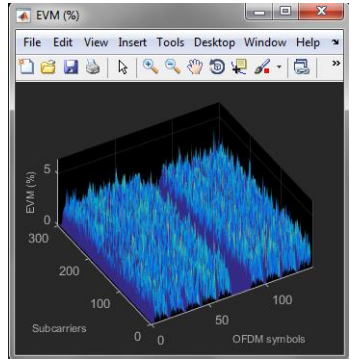


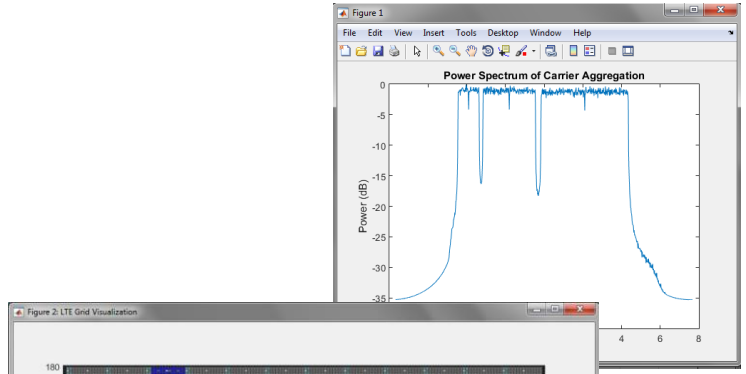
LTE System Toolbox

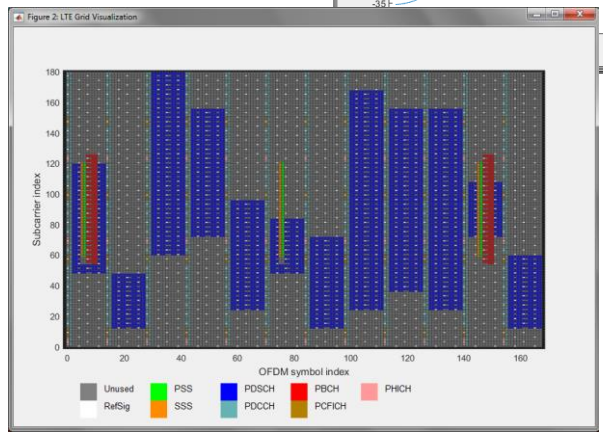
- LTE and LTE-Advanced (Rel-8 through **Rel-12**)
- Scope
  - FDD/TDD
  - Uplink/Downlink
  - Transmitter/Receiver



- ~200 functions for physical layer (PHY) modeling
- Signal generation for LTE & UMTS
- ACLR/EVM measurement
- Conformance Tests

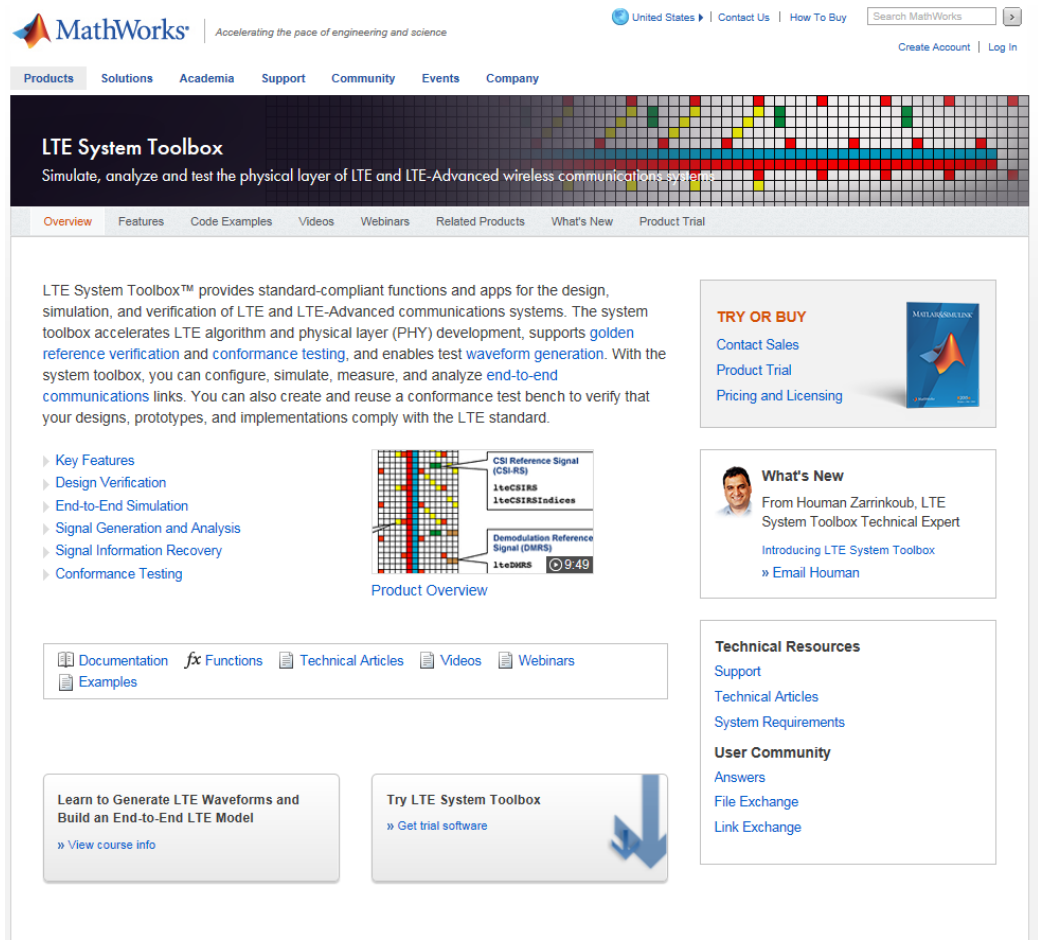






LTE System Toolbox | *More information…*

- Consult LTE Product Page
  - www.mathworks.com/products/lte-system/
  - Provides overview of LTE/LTE-A capabilities
  - Organized based on use-cases
- Consult Wireless Communications Page
  - www.mathworks.com/wireless
  - Provides overview of today's MATLAB® for Wireless System Design
- For details: Attend Recorded Webinar:
  - "Introducing LTE System Toolbox"



WLAN System Toolbox

- Physical layer (PHY) modeling

  Standard-compliant functions for the **design**, **simulation**, **analysis**, and testing of wireless LAN communications systems

- Transmitter & Receiver

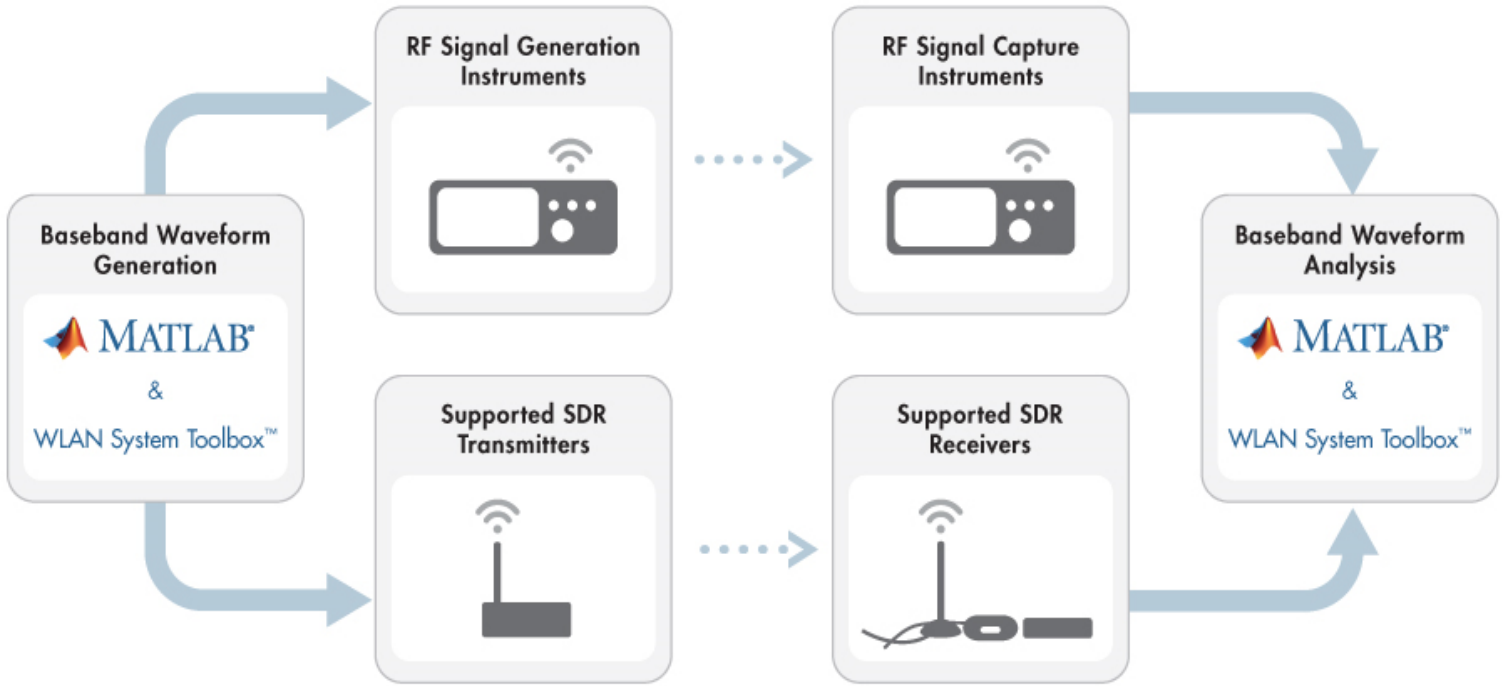
  L-SIG, HT-SIG, VHT-SIG-A, VHT-SIG-B
  OFDM, MIMO Equalization, STBC Combining
  Packet detection, symbol timing correction
  Coarse and fine frequency offset estimation
  Preamble signal decoders for L-SIG, HT-SIG, VHT-SIG-A, VHT-SIG-B fields

- Propagation Channel
  - TGn, TGac

- Measurements
  - Packet Error Rate, EVM, Spectral Emissions
- Features
  - Open, customizable MATLAB code
  - C-code generation with MATLAB Coder

WLAN System Toolbox

Hardware & Radio Connectivity



Generate custom waveforms

Transmit with SDR devices or RF instruments

Capture signals with SDR or instruments

Recover original data

Range of supported hardware

RF Signal Generator

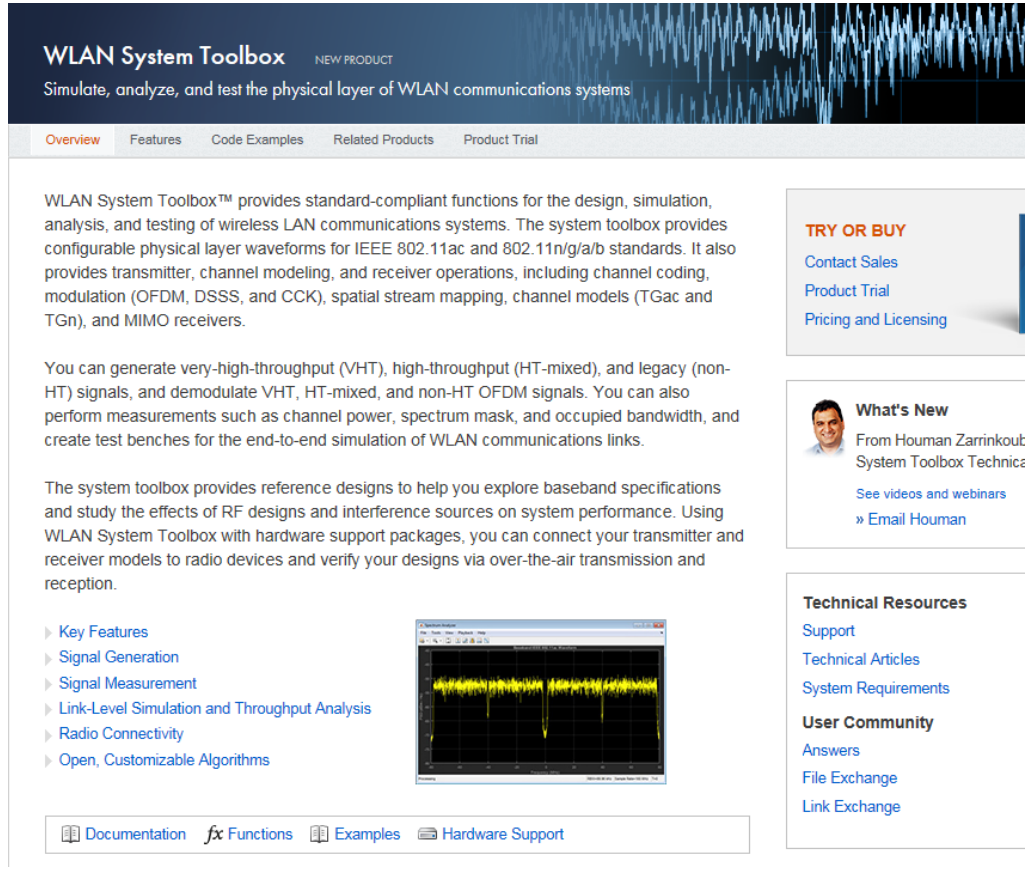
Spectrum Analyzer

Zynq Radio SDR

USRP SDR

WLAN System Toolbox | *More information…*

- Consult WLAN Product Page
  - www.mathworks.com/products/wlan-system/
  - Provides overview of WLAN capabilities
  - Organized based on use-cases
- Consult Wireless Communications Page
  - www.mathworks.com/wireless
  - Provides overview of today's MATLAB® for Wireless System Design
- For details: Attend Recorded Webinar:
  - "Introducing WLAN System Toolbox"



Signal Processing

Audio

Antenna to Bits

WLAN/LTE

Image and Video Processing

Image and Video Processing

- Stereo Camera Calibration R2014b
  - Lens distortion correction
  - Rectification
- Depth estimation R2014a
- 3D Scene reconstruction R2014a
- Code generation R2015a

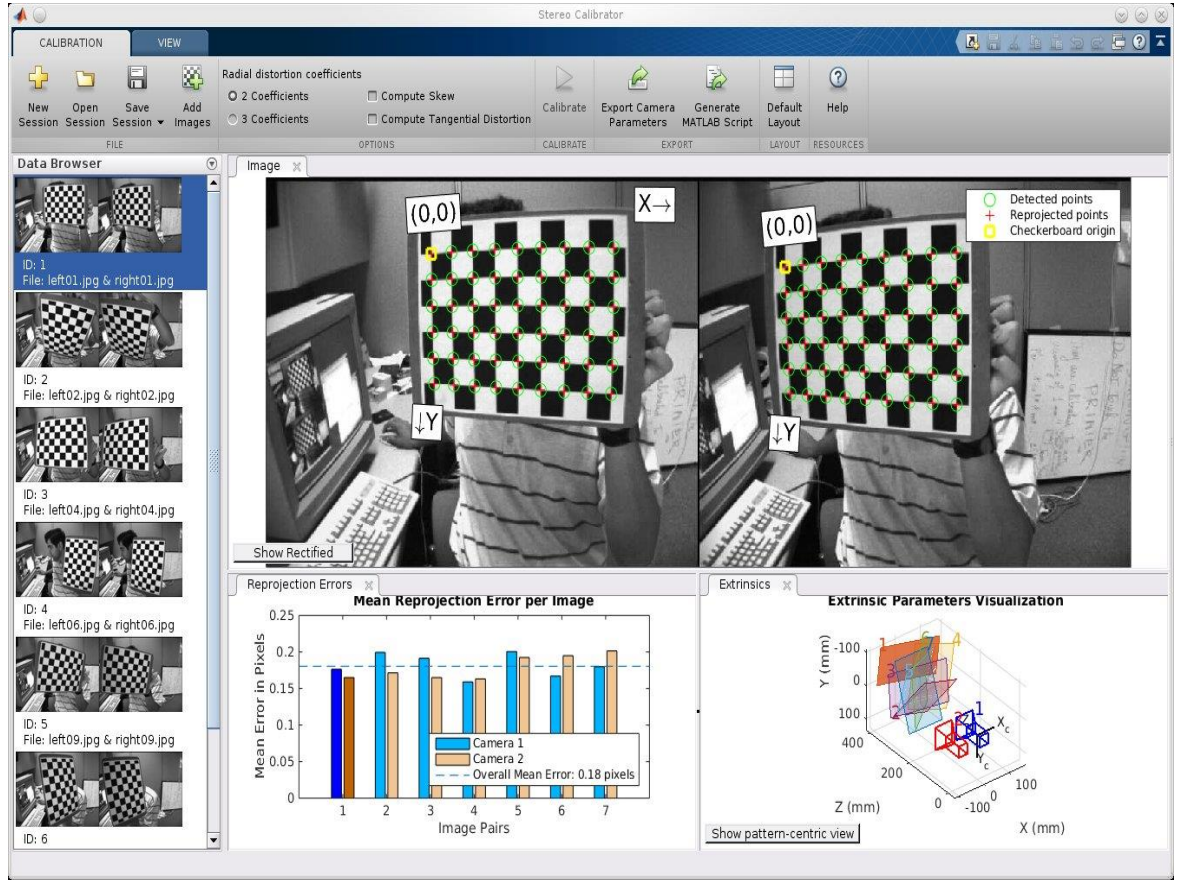


Image and Video Processing | *Stereo Vision*

R2016a

- Enables autonomous systems to **map and measure** the world
- Supports workflows for **ADAS**, autonomous driving, and robotics
- New functionality to support:
  - 3D **point cloud** processing
  - Structure from motion

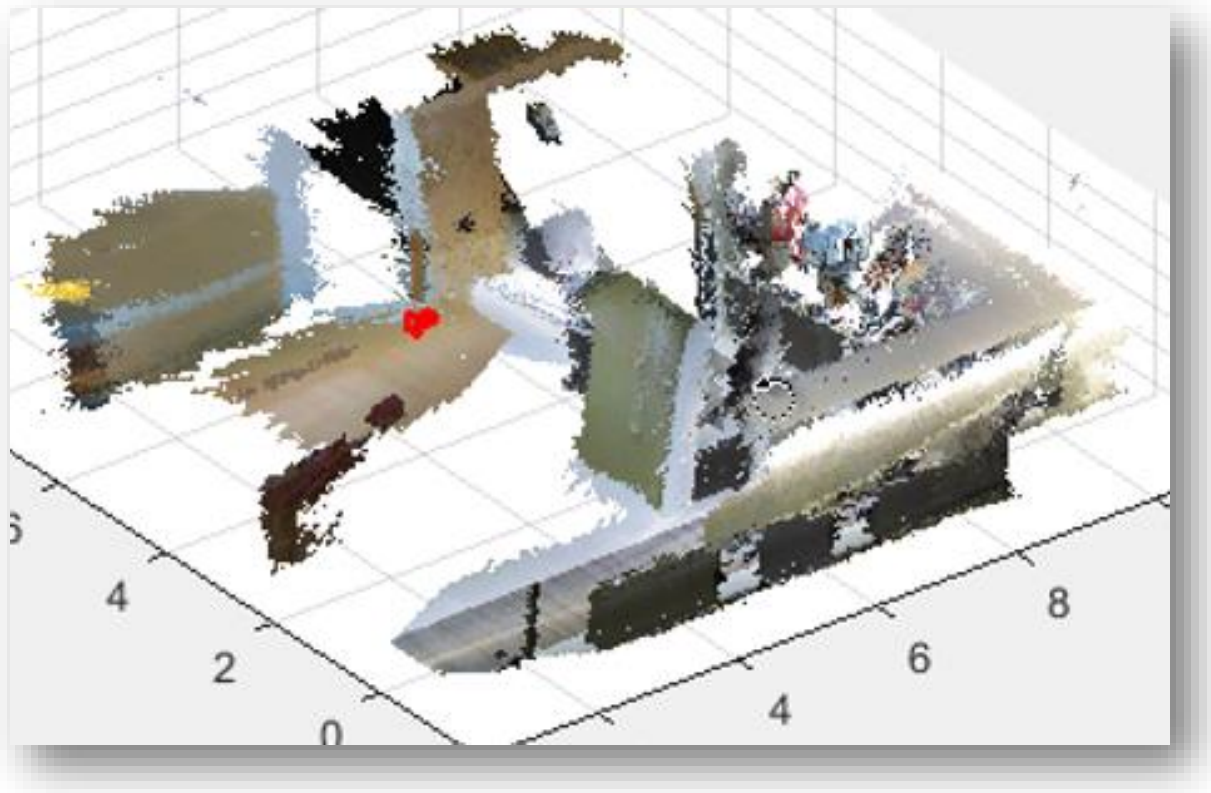
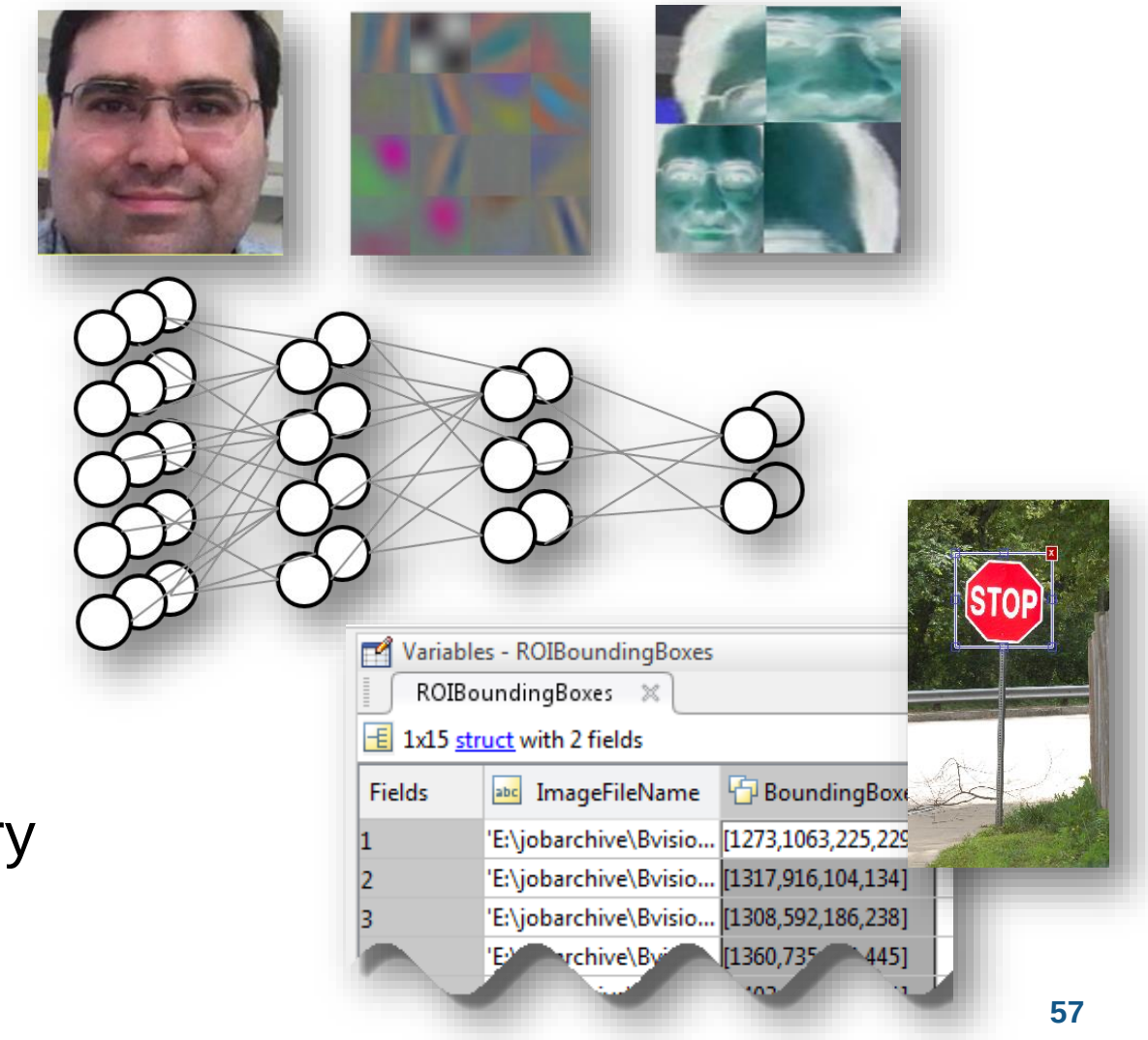


Image and Video Processing | *Deep Learning*

R2016a

- Perform fast, accurate image classification
- Enables recognition workflows in autonomous robotics and ADAS
- Convolutional neural network (CNN) algorithm added to Neural Network Toolbox
- Uses cuDNN (a GPU-accelerated library from NVIDIA)
  (requires Parallel Computing Toolbox)



- *Signal Processing*
- *Audio*
- *Antenna to Bits*
- *WLAN/LTE*
- *Image and Video Processing*

That’s, what’s new!