Challenges with Large Array Systems

- Design & simulation of multi-stage, multi-channel RF chains

- Large antenna arrays
  - Antennas need to be close together to avoid grating lobes
  - Digital beamforming can be complex and power hungry (BW x N_T, many ADCs)
  - Analog beamforming has limited capabilities

- Array structures are complex

- Design & simulation of multi-function, multi-domain systems
Agenda

- RF budget analysis and performance simulation of large arrays
- Partition beamforming between the digital and RF domains
- Antenna & array design
- Integrate antenna and array designs in system level models
- Summary
Project Requirements

- Requirements review
- Build large size transmit array models
- RF budget analysis and performance simulation
  - Gains of TX array and individual channels
  - Gain variations and array radiation pattern
  - Non-linearity via two-tone test
  - Phase noise and other RF impairments
Budget Analysis with RF Budget Analyzer
Demo: Build Large Size RF Transmit Array

Programmatically

RF Test Bench

RF Transmit Array

1:n split unit

Cascaded RF components
Specify the size of the array and click ‘Run’
Workflow (build large size transmit arrays)

- **Step 1:**
  - Build basic RF component chain models from an excel sheet
  - Introduce frequency dependent parameters, variations (randomness, e.g. gain), non-linearity, and other RF impairments, *if desired*
  - Modify them manually *if necessary* (*‘beautify’ the models*), and form a library of basic RF models (stage units)

- **Step 2:**
  - Build large size transmit array *programmatically* with basic RF models in the library and other Simulink and RF Blockset blocks

- **Step 3:**
  - Build test benches around the transmit array *programmatically*

Perform budget analysis and performance simulation
Examples

- **Step 1 example**
Examples

- Steps 2 & 3 combined example

```
34  % Specify top level system parameters
35  % Specify the size of the transmitter array
36  sizeArray = 64; % Has to be > 4 and be a power of 2;
37  %
38  filename = 'rfb_example.xlsx';
39  sheet = 'RF Component Chain';
40  GHz = 1e9;
41  InputFrequency = GHz * xlsread(filename,sheet,'B2');
42  SignalBandwidth = 10e6;
43  AvailableInputPower = 20;
44  %
45  % load the pre-built tx array model
46  d = buildTXArrayFun(sizeArray);
47  load_system('txArrayRF');
48  % Set a starting point in a blank model
49  x = 20;
50  dx = 40;
51  dy = 85;
52  v = 200 + dx*sizeArray/2;
```

Step 2
RF Budget Analysis and Performance Simulation

- Examine Gain/Power Levels
RF Budget Analysis and Performance Simulation

- Introduce gain variation & examine array radiation pattern
RF Budget Analysis and Performance Simulation

- Array radiation pattern and gain variation
RF Budget Analysis and Performance Simulation

- Examine non-linearity impact and introduce phase noise

Budget Analysis & Performance Simulation of Large Size Transmit Array

Sine Wave → Sum of Elements → dBm to Linear → Available input power (dBm)

Each tone : 20dBm/2
10*log10(A^2/2)=10-30;

Use what signals to push through? Depend on applications; non-single tone
What to observe? Spectrum contents, directivity; dynamic range, phase noise
Use spectrum scope?
Phase shift of components?
RF Budget Analysis and Performance Simulation

- Two-tone test (Non-Linearity Analysis) and phase noise
RF Budget Analysis and Performance Simulation

- Two-tone test and phase noise
Project Requirements- Workflow Solution

- Export the basic RF channel built from an Excel spreadsheet in RF Budget Analyzer into Simulink/RF Blockset; Introduce the desired RF impairments into the model

- Build a library of basic RF units from the single RF channel Simulink/RF Blockset model; Form multiple staged large size arrays from basic RF units programmatically

Further requirements
- Add power saturation for amplifiers
- Add power efficiency metric
- Add frequency dependency to the arrays
## RF Budget Analyzer vs. RF Blockset

<table>
<thead>
<tr>
<th>Table Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytical calculation vs. numerical simulation</td>
<td></td>
</tr>
<tr>
<td>Cascaded configuration vs. arbitrary topology</td>
<td></td>
</tr>
<tr>
<td>Formulas vs. dynamic multi-domain simulation</td>
<td>(circuit simulator using circuit envelope technology)</td>
</tr>
<tr>
<td></td>
<td>(quantization noise, non-linearity, thermal and phase noise, and other RF impairments)</td>
</tr>
</tbody>
</table>
Partition beamforming between the digital and RF domains
Challenges Designing Massive MIMO Arrays for Systems

- Higher frequencies enable more antennas
  - mmWave band (28 GHz, 37 GHz, etc…)
  - Large number of antennas, 32, 64, ….

- Large antenna arrays
  - Needed to provide more beamforming gain to overcome the path loss
  - T/R module is needed behind each element
  - Architecture is difficult to build due to cost, space, and power limitations
What is Hybrid Beamforming?

- Beamforming implemented part in the digital and part in the RF domain
  - Trade-off performance, power dissipation, implementation complexity
- Subarrays contain RF channels with phase shifter
- Digital beamforming performed on signals outside subarrays
Example: System Architecture for Hybrid Beamforming

- The transmitter uses a larger array to perform beamforming towards the receiver.
- The receiver estimates the direction of arrival with small orthogonal arrays and communicates it to the transmitter.
Example: Hybrid Beamforming Transmitter Array

- 4 subarrays of 8 patch antennas operating at 66GHz → 8x4 = 32 antennas
- Digital beamforming applied to the 4 subarrays (azimuth steering)
- RF beamforming (phase shifters) applied to the 8 antennas (elevation steering)
RF Front End Modelling using Circuit Envelope

- Direct conversion to IF (5GHz) and superhet up-conversion to mmWave (66GHz)
- Non-linearity (e.g. IP2, IP3, P1dB)
- Power dividers (e.g. S-parameters)
- Variable phase-shifters
Antenna and Array Design
Easier Antenna Design with Antenna Toolbox

- Design is easy and natural
  - Library of parameterized antenna elements
  - Functionality for the design of antenna arrays
  - CAD description streamlined
- Rapid simulation setup
  - Full Methods of Moments solver employed for ports, fields 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
Building your First Antenna and Antenna Array

```matlab
p = patchMicrostrip
p.Height = 0.01;
impedance(p, (500e6:10e6:2e9));
current(p, 1.7e9);
pattern(p, 1.7e9);

a = linearArray
a.Element = p;
a.ElementSpacing = 0.1;
a.NumElements = 4;
show(a);
patternElevation(a, 1.7e9, 0);
```
What if my Antenna is not in the Library?

- Define the boundary of your custom planar (2D) structure
  - Basic shapes: rectangle, circle, polygon
  - Operations: intersection, union, difference
- Define the feeding point (inset or probe)
- Integrate your custom antenna
  - Define a backing structure
  - Define a dielectric structure
  - Build an array with custom elements

```matlab
plate = antenna.Rectangle('Length',0.16,'Width',0.16);
notch1 = antenna.Circle('Center',[0,0.06],'Radius',.06);
notch2 = antenna.Rectangle('Length',0.15,'Width',.005);
b = plate-notch1-notch2;
```
What if I Need to Customize my Array?

- Build regular arrays where you can change the properties of individual elements (rotation, size, tapering)
  - Linear, Rectangular, Circular array
- Describe conformal (heterogeneous) arrays in terms of element type and arbitrary position
  - Conformal array (both balanced and unbalanced)
- Arbitrary shape designed with custom geometry or mesh

```matlab
arr = conformalArray;
d = dipole;
b = bowtieTriangular;
arr.Element = {d, b};
arr.ElementPosition(1,:) = [0 0 0];
arr.ElementPosition(2,:) = [0 0.5 0];
```
What if my Array is Really Large?

- Infinite Array Analysis
  - Repeat unit cell (Same Element) infinitely
  - Impedance and pattern become function of frequency and scan angle
  - Ignore edge effects
  - Captures mutual coupling
- Validate with full wave simulation on smaller arrays

Scan Impedance @10GHz

0deg Azimuth 45deg Azimuth 90deg Azimuth

Scan Impedance
0deg Azimuth 45deg Elevation

Power Pattern
Increasing the Efficiency of the Antenna Design Workflow

Modelling the dielectric substrate can slow down analysis time:

- Use antennas in free space for first-cut design
  - Combine with optimization routines to rapidly find out a suitable starting point
- Use parallel computing to speed up design space exploration

```matlab
patternoptions = psoptimset(@patternsearch);
patternoptions.PlotFons = @psplotbestf;
patternoptions.MaxIter = 25;

optimdesign = patternsearch(@(x) yagi_objective_function(yagidesign,x,freq,elang),...;
    parasitic_values,[],[],[],[],LB,UB,[],patternoptions);
```

![Poor directivity](image1)
![Optimized pattern](image2)

Speed-up due to parallel computing = 4.80134

<table>
<thead>
<tr>
<th>Without Parallel Computing</th>
<th>With Parallel Computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>113.3</td>
<td>23.597</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
</tr>
</tbody>
</table>
Array Synthesis from a Desired Pattern
Array Synthesis from a Desired Pattern

1. Desired Beam Pattern
2. Initial Beam Pattern
3. Develop Cost Function to Minimize the Difference Between the Desired and the Resulting Patterns
4. Run Through Optimization
5. Generate Weights (and optionally element positions) to produce the pattern

%%% Optimize pattern using only weights

```
N = 8; % 8 elements in a linear array
azimuth = -90:90; % Define azimuth field of view

weights_d = hamming(N); % Create weights for desired pattern
weights_d = weights_d/norm(weights_d); % Normalize weights for pattern

stvmat = steervec((0:N-1)/2,azimuth); % Generate a pattern for this example
Beam_d = abs(weights_d' * stvmat); % Apply weights to desired pattern
```
Results After Optimization

%% Set up optimization

objfun = @(w)norm(abs(w'*stvmat)-Beam_d);

weights_i = ones(N,1);

weights_o = fmincon(objfun,weights_i,[]);
Integration of Antenna Array with Spatial Signal Processing Algorithms
Combine Antenna Design and Phased Array Algorithms

- You can integrate your antenna in Phased Array System Toolbox array objects
  - Use the accurate far field (complex) radiation pattern of the antenna
- Phased Array System Toolbox provides algorithms and tools to design, simulate, and analyze phased array signal processing systems
  - Beamforming, Estimation of Direction of Arrival
- Uses pattern superposition to compute the array pattern

...% Import antenna element in Phased Array
myantenna = dipole;
myURA = phased.URA;
myURA.Element = myantenna;
Accelerate Algorithm Execution

- Use Best Practices in Programming
  - Vectorization
  - Pre-allocation
- Parallel Computing
  - High level parallel constructs (e.g. `parfor`)
  - Utilize cluster, clouds, and grids
- MATLAB to C
- GPUs
MATLAB & Simulink: Unified Design Platform
for baseband, RF, and antenna modeling and simulation

Algorithms, Waveforms, Measurements
- Communications System Toolbox
- Phased Array System Toolbox
- LTE System Toolbox
- WLAN System Toolbox

RF Front End
- RF Toolbox
- RF Blockset

Antennas, Antenna Arrays
- Antenna Toolbox
- Phased Array System Toolbox

System Architecture
- DSP Algorithms
- Mixed-Signal
- RF Design
- Antenna Design

Baseband
- Digital Front End

Digital PHY
- Baseband
- Digital Front End

RF Front End
- DAC
- PA
- ADC
- LNA

Mixed-signal
- Simulink
- DSP System Toolbox
- Control System Toolbox

Channel Modeling
- Communications System Toolbox
- Phased Array System Toolbox
- LTE System Toolbox
- WLAN System Toolbox

Channel
What's new in R2017a?
Antenna Design – Where To Start?

Antenna Designer App

- Select an antenna based on the desired specifications
- Design the antenna at the operating frequency
- Visualize results and iterate on antenna geometrical properties
- Generates MATLAB scripts for automation
Coverage and Field Strength Visualization on Map

- Compute antenna pattern and visualize field strength projected on flat earth map
- Visualize antenna coverage on flat earth map and communication links
  - Define transmitter and receiver
  - Antenna design, frequency, power, and sensitivity
What’s new in Phased Array System Toolbox

5G Beamforming and Scatterer MIMO Channel

Active and Passive Sonar

Range and Doppler Estimation

New Features, Compatibility Considerations

- Scattering MIMO Channel: Model multipath signal propagation through spatially spread scatterers
- Sonar Systems: Model hydrophones, projectors, underwater propagation, and targets
- Range and Doppler Estimation: Measure target range and speed
5G Beamforming and Spatial MIMO Channel

Scatterer MIMO Channel Model
- Generic model, applicable to all 5G bands and array sizes
- Multipath due to single reflection from multiple scatterers

Diagonalization Beamformer
- Precoding and combining weights
- Power distribution using water-filling algorithm
- Subchannel gains and channel capacity estimation

Examples
- Antenna Arrays in MIMO Communications
- MIMO-OFDM Precoding with Phased Arrays \textit{(with CST)}
- 802.11ad Waveform Generation with Beamforming \textit{(with WST)}
Active and Passive Sonar Systems

Sonar Arrays and Targets
- Hydrophones
- Projectors
- Backscatter sonar target

Underwater Channel Model
- Isospeed

Examples
- Locating an Acoustic Beacon with a Passive Sonar
- Underwater Target Detection with an Active Sonar

1 incl. integration with BELLHOP from HLS Research’s Acoustic Toolbox
Summary:

- Trusted, diverse set of libraries and algorithms
- Fast simulations with scalable computing across CPU, GPU, and Clusters
- Unified modelling and simulation of digital, RF, and antenna systems
- Integrated platform for mathematical analysis, and algorithm, software, & hardware development
Call to Action

- Download whitepapers, technical articles and watch recorded webinars

- Webinar: Design of wireless MIMO systems: from RF specifications to architecture exploration
- Design and Verify RF Transceivers for Radar Systems
- Wideband Radar System Design
- Designing Antennas and Antenna Arrays with MATLAB and Antenna Toolbox
- Hybrid Beamforming for Massive MIMO Phased Array Systems
- Synthesizing an Array from a Specified Pattern: An Optimization Workflow
Do You Want To Learn More?
Phased Array System Toolbox Fundamentals

This one-day course provides a comprehensive introduction to the Phased Array System Toolbox™. Themes including radar characterization and analysis, radar design and modeling and radar signal processing are explored throughout the course.

Topics include:

- Review of a Monostatic End-to-End Radar Model
- Characterize and analyze radar components and systems
- Design and model components of a radar system
- Implement a range of radar signal processing algorithms
Topics include:

- Introduction to RF simulation using MathWorks tools
- How do I model my RF system with RF Blockset?
- Importing S-Parameters and modeling linear operation
- Fundamentals of noise simulation
- Modeling non-linear devices
- Developing custom models
Speaker Details
Email: tabrez.khan@mathworks.in

Contact MathWorks India
Products/Training Enquiry Booth
Call: 080-6632-6000
Email: info@mathworks.in

Your feedback is valued.
Please complete the feedback form provided to you.
Thanks for your attention

Questions?