Design and Simulation of Communications and Radar Systems

Armerkom
Istanbul, Turkey
February 28, 2014

Nick M. Zayed
# Agenda

<table>
<thead>
<tr>
<th>Time</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00 – 10:00</td>
<td><strong>Building Communication System Simulations</strong></td>
</tr>
<tr>
<td></td>
<td>• Constructing a basic QPSK communications model in Simulink</td>
</tr>
<tr>
<td></td>
<td>• 802.11a Wireless LAN example</td>
</tr>
<tr>
<td></td>
<td>• Adding RF design detail</td>
</tr>
<tr>
<td>10:00 – 10:45</td>
<td><strong>4G LTE Design using the LTE System Toolbox</strong></td>
</tr>
<tr>
<td>10:45 – 11:00</td>
<td>Break</td>
</tr>
<tr>
<td>11:00 – 11:30</td>
<td><strong>Options for Accelerating MATLAB Simulations</strong></td>
</tr>
<tr>
<td>11:30 – 12:15</td>
<td><strong>Electronic Interference and Array Signal Processing</strong></td>
</tr>
<tr>
<td>12:15 – 13:15</td>
<td>Lunch</td>
</tr>
<tr>
<td>13:15 – 14:15</td>
<td><strong>End-to-End Modeling of Radar Systems</strong></td>
</tr>
<tr>
<td></td>
<td>• Radar Analysis with User-Interface Apps</td>
</tr>
<tr>
<td></td>
<td>• Integrating the RF Subsystem</td>
</tr>
<tr>
<td>14:15 – 14:30</td>
<td>Break</td>
</tr>
<tr>
<td>14:30 – 16:30</td>
<td><strong>Hands-on Workshop: Phased Array Systems</strong></td>
</tr>
</tbody>
</table>
Building Communication System Simulations

9:00 – 10:00
Challenges: From specification to implementation

- Elaborate specifications into models
- Assess system-level performance of your design
- Introduce innovative proprietary algorithms
Where do MATLAB and Simulink fit?

- MATLAB and Simulink for algorithm development and analysis
  - Floating and Fixed-Point

- MATLAB and Simulink for dynamic and large scale system simulations
  - Uniting Baseband and RF

- Quickly proceed from MATLAB and Simulink models to implementation with…
  - C and HDL code generation
Core MathWorks Products

**SIMULINK®**

The leading environment for modeling, simulating, and implementing dynamic and embedded systems

- Linear & nonlinear, continuous & discrete-time, hybrid, and multi-rate systems
- Foundation for model-based design, including physical-domain modeling, automatic code generation, and verification and validation
- Open architecture for integrating models from other tools
- Applications in controls, signal processing, communications, and other system engineering areas
Communications System Toolbox

Over 100 algorithms for
- Modulation, Interleaving, Channels, Source Coding
- Error Coding and Correction
- MIMO, Equalizers, Synchronization
- GPU-optimized components, SDR hardware

Algorithm libraries in MATLAB

Algorithm libraries in Simulink
Building a Physical Layer (PHY) model of a QPSK communications system
IEEE 802.11a Wireless LAN System

Demo
>> wlan_eml_noRF
Communications System Toolbox

Communications System Toolbox™ provides algorithms and tools for the design, simulation, and analysis of communications systems.
Sharing information between digital and RF engineers

Here’s my digital model. Can you add some RF?

How much RF would you like?

Noise figure, IP3?

Measured S-parameters?

Circuit level design?

Idealized baseband

Equalized baseband

RF Engineer

Communication System Engineer
RF Modeling Technologies in Simulink

Circuit envelope
- Multi-carrier systems and arbitrary architectures

Equivalent baseband
- Single carrier cascaded systems

Idealized baseband
- Mathematical analytical models

Demo
>> simRF libraries
>> wlan_eml_rf3
Direct Conversion ISM Band Receiver: Circuit Envelope Model

Demo
>> simrfV2_direct_conv
Summary

- Simulink is the ideal platform for system-level simulation
  - Works seamlessly with MATLAB
  - Provides multi-domain modeling capability (RF, Analog, and Digital)

- Communications System Toolbox is ideal for getting baseband models up and running quickly

- SimRF provides an easy-to-use way to include RF effects in system-level simulations
# Agenda

<table>
<thead>
<tr>
<th>Time</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00 – 10:00</td>
<td><strong>Building Communication System Simulations</strong></td>
</tr>
<tr>
<td></td>
<td>• Constructing a basic QPSK communications model in Simulink</td>
</tr>
<tr>
<td></td>
<td>• 802.11a Wireless LAN example</td>
</tr>
<tr>
<td></td>
<td>• Adding RF design detail</td>
</tr>
<tr>
<td>10:00 – 10:45</td>
<td><strong>4G LTE Design using the LTE System Toolbox</strong></td>
</tr>
<tr>
<td>10:45 – 11:00</td>
<td>Break</td>
</tr>
<tr>
<td>11:00 – 11:30</td>
<td><strong>Options for Accelerating MATLAB Simulations</strong></td>
</tr>
<tr>
<td>11:30 – 12:15</td>
<td><strong>Electronic Interference and Array Signal Processing</strong></td>
</tr>
<tr>
<td>12:15 – 13:15</td>
<td>Lunch</td>
</tr>
<tr>
<td>13:15 – 14:15</td>
<td><strong>End-to-End Modeling of Radar Systems</strong></td>
</tr>
<tr>
<td></td>
<td>• Radar Analysis with User-Interface Apps</td>
</tr>
<tr>
<td></td>
<td>• Integrating the RF Subsystem</td>
</tr>
<tr>
<td>14:15 – 14:30</td>
<td>Break</td>
</tr>
<tr>
<td>14:30 – 16:30</td>
<td><strong>Hands-on Workshop: Phased Array Systems</strong></td>
</tr>
</tbody>
</table>
What is the LTE System Toolbox?

- Release 8,9 and 10 (LTE-A)
- ~200 functions for physical layer (PHY) modeling
- Link-level simulation
  - No C or HDL code generation
- Scope
  - TDD & FDD,
  - Uplink & Downlink
  - Transmitter & Receiver
- Conformance tests
Standard-Compliant Solution

- Tested against instrument-generated signals
  - Rohde & Schwarz
  - Agilent

- Used in the industry since 2009
  - Initially under Steepest Ascent
  - MathWorks acquired Steepest Ascent

- Successfully demodulates live LTE signals captured in the field
Typical Use Cases

1) Golden reference to verify in-house PHY models

My PDSCH == ? LTE PDSCH

2) Modular end-to-end link-level simulation

3) Signal generation and analysis

4) Signal information recovery
"We used LTE System Toolbox to validate our LTE baseband IP cores and example design C models. This meant we could trust our own C models matched the 3GPP specification and allowed us to verify our RTL designs against the C models with confidence. **Catching issues early in simulation saved a lot of time later when it came to hardware testing.** The MATLAB environment allowed us to rapidly assemble and run test cases."

Bill Wilkie, Director of Communications Signal Processing (Europe), Xilinx.
(1) Golden Reference for Verification

- **Objective:**
  - Need to validate internal models against an external reference

- **Features:**
  - Standard-compliant reference
  - Varying levels of modeling detail:
    - low = scrambling, turbo coder, …
    - medium = transport channels, physical channels

- **Benefits:**
  - Reduce risk
  - Detect errors early
  - Independently confirm understanding of specifications
"We rely on Steepest Ascent's LTE Toolbox for a variety of project needs. One use is as a golden reference for our own in-house simulator where the Toolbox's complete feature set has allowed us to build industrial strength simulations rapidly with guaranteed baseline requirements. Its extensible design enables us to customise these models for our own LTE research work. All this is made easier by the great documentation and excellent product support."
(2) End-to-End Link-Level Simulation

**Objective:**
- Want to design or optimize a subset of the LTE system

**Features:**
- Complete, verified environment available
- Simulations already set up to measure link-level performance
  - Throughput, BER, block error rate

**Benefits:**
- Time savings
- Easy maintenance (documentation, support & tracks standard)
- You can focus on added value
(2) Demo: Equalizing the Downlink Grid

Transmitter
- Test Waveform Generation

Channel
- Fading Channel

Receiver
- Synchronisation & OFDM Demodulation
- Channel Estimation & Equalisation

Figure 1: LTE Grid Visualization

- OFDM symbol index
- Subcarrier index
- Unused, PSS, SSS, PDSCH, PBCH, PHICH, PCFICH
- Received resource grid
- Equalized resource grid
- Symbol
(3) Signal Generation and Analysis

- **Objective:**
  - Need to generate standard-compliant signals, for example to test a component such as an antenna or an amplifier

- **Features:**
  - High-level tools that generate customizable standard-compliant signals
    (lteRMCUlTool, lteRMCuDLTool, lteTestModelTool)
  - Powerful analysis capabilities (EVM, ACLR, …)

- **Benefits:**
  - No need for LTE expertise
  - Flexibility due to parameters and MATLAB code
  - Realistic signals to test components and model interferences
(3) Signal Generation and Analysis

Reference Measurement Channels

TS 36.101

Standard-compliant signal available in the MATLAB workspace

>> ItRMCDLTool
We have used Steepest Ascent software to successfully process the standards-based aspects of data produced from live exercises. Based on the excellent performance of the toolkit, we feel that our decision was correct and justified.
(4) Signal Information Recovery

- **Objective:**
  - Want to extract information from a real-world LTE signal

- **Features:**
  - Synchronization and demodulation of actual signal
  - Signal intelligence such as MIB and SIB1 extraction
  - Positioning information

- **Benefits:**
  - Off-the-shelf demodulation capability
  - Extensible, MATLAB-based solution

>> SIB1RecoveryExample
Ease of Use and Flexibility

- Granular transmit receive processing chains provide access to

- Ready-to-use conformance signals
- Transport-channel level functions
- Detail of processing chain
High Level: Conformance Signals

Signal Generators

GUI-based

MATLAB Function
Mid-Level: Transport Channels

Downlink overview

Uplink example
Fine: Details of Processing Chain

PDSCH Example

- Scrambling
  - ItcPDSCHPRBS
- Modulation
  - ItcSymbolModulate
  - ItcSymbolDemodulate
- Demodulation
- Layer Mapping
  - LteLayerMap
  - LteLayerDemap
- Precoding
  - LteDLPrecode
  - LteLayerDeprecode

Complete PDSCH processing:
- Encoding: ItcPDSCH
- Decoding: ItcPDSCHDecode

Resource indices:
- ItcPDSCHIndices

Precoding Matrix
- Indication
  - ItcPMISelect
  - ItcPMInfo
  - ItcCSICodebook

Fine
Mid-level
### LTE Toolbox channel support

- Comprehensive set of functions covering:

<table>
<thead>
<tr>
<th>Downlink</th>
<th>Uplink</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transport Channels &amp; Control Information</strong></td>
<td><strong>Transport Channels &amp; Control Information</strong></td>
</tr>
<tr>
<td>DL-SCH</td>
<td>HI</td>
</tr>
<tr>
<td>BCH</td>
<td>CFI</td>
</tr>
<tr>
<td>DCI</td>
<td></td>
</tr>
<tr>
<td><strong>Physical Channels and Signals</strong></td>
<td><strong>Physical Channels and Signals</strong></td>
</tr>
<tr>
<td>PDSCH</td>
<td>PDCCH</td>
</tr>
<tr>
<td>PBCH</td>
<td>Reference Signals</td>
</tr>
<tr>
<td>PCFICH</td>
<td>PSS - SSS</td>
</tr>
<tr>
<td>PHICH</td>
<td></td>
</tr>
</tbody>
</table>
Documentation

Shipping Examples

LTE System Toolbox Examples

On this page...
- Downlink LTE Modeling
- Uplink LTE Modeling
- Downlink End to End Simulation
- Uplink End to End Simulation
- Downlink Waveform Generation and Analysis
- Uplink Waveform Generation and Analysis

Downlink LTE Modeling

LTE Waveform Modeling Using Downlink Transport and Physical Channels

PDSCH Transmit Diversity Throughput Simulation

PDSCH Port 5 UE-Specific Beamforming

Release 10 PDSCH Enhanced UE-Specific Beamforming

Functions

Functions in LTE System Toolbox

LTE Modeling Basics
- lteResourceGrid
- lteResourceGridSize
- lteDLResourceGrid
- lteDLResourceGridSize
- lteULResourceGrid
- lteULResourceGridSize
- lteDuplexingInfo

Subframe resource array
Size of subframe resource array
Downlink subframe resource array
Size of downlink subframe resource array
Uplink subframe resource array
Size of uplink subframe resource array
Duplexing information

Downlink Channels

Physical Signals
- ltePSS
- ltePSSIndices
- lteSSS
- lteSSSIndices
- lteCellRS
- lteCellRSSIndices

Primary synchronization signal
PSS resource element indices
Secondary synchronization signal
SSS resource element indices
Cell-specific reference signal
CRS resource element indices
Summary of Main Benefits

- Comprehensive
  - Comprehensive set of PHY models
  - Numerous preset, extensible examples

- Open environment
  - MATLAB-based
  - Link to test and measurement instruments

- Standard-compliance
  - Tested against hardware-based signal generators
  - Trusted by numerous customers
## Agenda

<table>
<thead>
<tr>
<th>Time</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00 – 10:00</td>
<td><strong>Building Communication System Simulations</strong></td>
</tr>
<tr>
<td></td>
<td>• Constructing a basic QPSK communications model in Simulink</td>
</tr>
<tr>
<td></td>
<td>• 802.11a Wireless LAN example</td>
</tr>
<tr>
<td></td>
<td>• Adding RF design detail</td>
</tr>
<tr>
<td>10:00 – 10:45</td>
<td><strong>4G LTE Design using the LTE System Toolbox</strong></td>
</tr>
<tr>
<td>10:45 – 11:00</td>
<td><strong>Break</strong></td>
</tr>
<tr>
<td>11:00 – 11:30</td>
<td><strong>Options for Accelerating MATLAB Simulations</strong></td>
</tr>
<tr>
<td>11:30 – 12:15</td>
<td><strong>Electronic Interference and Array Signal Processing</strong></td>
</tr>
<tr>
<td>12:15 – 13:15</td>
<td><strong>Lunch</strong></td>
</tr>
<tr>
<td>13:15 – 14:15</td>
<td><strong>End-to-End Modeling of Radar Systems</strong></td>
</tr>
<tr>
<td></td>
<td>• Radar Analysis with User-Interface Apps</td>
</tr>
<tr>
<td></td>
<td>• Integrating the RF Subsystem</td>
</tr>
<tr>
<td>14:15 – 14:30</td>
<td><strong>Break</strong></td>
</tr>
<tr>
<td>14:30 – 16:30</td>
<td><strong>Hands-on Workshop: Phased Array Systems</strong></td>
</tr>
</tbody>
</table>
Options for Accelerating MATLAB Simulations

11:00 – 11:30
Simulation acceleration options in MATLAB

Best Practices

MATLAB to C (MEX)

Parallel Computing

GPU processing

Pre-allocation / Vectorization System Objects

MATLAB Coder

Parallel Computing Toolbox

GPU System objects
- comm.gpu.TurboDecoder
- comm.gpu.ViterbiDecoder
- comm.gpu.LDPCDecoder
- comm.gpu.PSKDemodulator
- comm.gpu.AWGNChannel
- phased.gpu.ConstantGammaClutter

User’s Code

for k=1:max
    x = fft(data)
    y = 20*log10
Parallel Simulation Runs

- Matlab
- Simulink
- Toolboxes
- Blocksets

Time

- Worker
- Worker
- Worker

Tasks:
- Task 1
- Task 2
- Task 3
- Task 4
parfor construct

- `matlabpool` → available workers
- No modification of algorithm
- **Use `parfor` loop instead of `for` loop**
- Parallel computation leads to further acceleration
- More cores = more speed

```matlab
tic;
parfor param=1:MaxSNR
    fprintf(1,'Iteration number %d\n',param);
    y=zTransceiver_v5(x,param);
end
f=toc;
```

Demos
>> DemoScript_SimAccel
# MATLAB Acceleration Strategies

<table>
<thead>
<tr>
<th>Option</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. <strong>Best Practices in Programming</strong></td>
<td>MATLAB, Toolboxes, System Toolboxes</td>
</tr>
<tr>
<td>• Vectorize &amp; Pre-allocate Arrays &amp; Matrices</td>
<td></td>
</tr>
<tr>
<td>• Use System Objects instead of functions</td>
<td></td>
</tr>
<tr>
<td>2. <strong>Compiled Code (MEX)</strong></td>
<td>MATLAB, MATLAB Coder</td>
</tr>
<tr>
<td>3. <strong>More Processors or Cores</strong></td>
<td>Parallel Computing Toolbox</td>
</tr>
<tr>
<td>• Parallel constructs (e.g., <code>parfor</code>, <code>matlabpool</code>)</td>
<td></td>
</tr>
<tr>
<td>• GPUs (GPU system objects)</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>Title</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------</td>
</tr>
<tr>
<td>9:00 – 10:00</td>
<td><strong>Building Communication System Simulations</strong></td>
</tr>
<tr>
<td></td>
<td>• Constructing a basic QPSK communications model in Simulink</td>
</tr>
<tr>
<td></td>
<td>• 802.11a Wireless LAN example</td>
</tr>
<tr>
<td></td>
<td>• Adding RF design detail</td>
</tr>
<tr>
<td>10:00 – 10:45</td>
<td><strong>4G LTE Design using the LTE System Toolbox</strong></td>
</tr>
<tr>
<td>10:45 – 11:00</td>
<td>Break</td>
</tr>
<tr>
<td>11:00 – 11:30</td>
<td><strong>Options for Accelerating MATLAB Simulations</strong></td>
</tr>
<tr>
<td>11:30 – 12:15</td>
<td><strong>Electronic Interference and Array Signal Processing</strong></td>
</tr>
<tr>
<td>12:15 – 13:15</td>
<td>Lunch</td>
</tr>
<tr>
<td>13:15 – 14:15</td>
<td><strong>End-to-End Modeling of Radar Systems</strong></td>
</tr>
<tr>
<td></td>
<td>• Radar Analysis with User-Interface Apps</td>
</tr>
<tr>
<td></td>
<td>• Integrating the RF Subsystem</td>
</tr>
<tr>
<td>14:15 – 14:30</td>
<td>Break</td>
</tr>
<tr>
<td>14:30 – 16:30</td>
<td><strong>Hands-on Workshop: Phased Array Systems</strong></td>
</tr>
</tbody>
</table>
Electronic Interference and Array Signal Processing

11:30 – 12:15
Modeling Electronic Interference Scenarios

- Partition model into Transmitter, Jammer, and RF subsystems
- Introduce Jamming, Co-channel Interference, and Smart Antenna Configurations
- Simulate to determine expected performance and operational limits
Array Signal Processing Algorithms
Receiving Signals with an Omnidirectional Antenna
Receiving Signals with a Sensor Array
Receiving a Broadside Signal (azimuth = 0) with a Sensor Array

$\Delta t = 0$
Aligned or Coherent Signals

Signal Wavefronts

Enhanced Signal (coherent sum)
Receiving a Signal from an Arbitrary Azimuth ($\theta$)

$$\Delta t = \frac{d}{c} \sin \theta$$
Non-aligned or Non-coherent Signals

Signal Wavefronts

$t_1 = \Delta t$
$t_2 = 2\Delta t$

$\Delta t = \frac{d}{c} \sin \theta$
Array Response Pattern
Alignment via time delays

Signal Wavefronts

\[ t_1 \quad \text{delays} \quad t_2 \]

Aligned Signals

\[ \Sigma \]

Enhanced Signal (coherent sum)
Converting Time Delays to Carrier Phase Shifts

\[ \Delta t = \frac{d}{c} \sin \theta \]
\[ \Delta \phi = 2\pi f \Delta t \]
\[ \Delta \phi = 2\pi f \frac{d}{c} \sin \theta \]
\[ \lambda = \frac{c}{f} \]
\[ d = \lambda / 2 \]
\[ \Delta \phi = \pi \sin \theta \]
Electronic Steering

Signal Wavefronts

Phase Shift

Aligned Signals

Enhanced Signal (coherent sum)

$\phi_n = A + iB$

$\Delta \phi = \pi \sin \theta$

Demos
>> Array Processing
<table>
<thead>
<tr>
<th>Time</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00 – 10:00</td>
<td><strong>Building Communication System Simulations</strong>&lt;br&gt;• Constructing a basic QPSK communications model in Simulink&lt;br&gt;• 802.11a Wireless LAN example&lt;br&gt;• Adding RF design detail</td>
</tr>
<tr>
<td>10:00 – 10:45</td>
<td><strong>4G LTE Design using the LTE System Toolbox</strong></td>
</tr>
<tr>
<td>10:45 – 11:00</td>
<td><strong>Break</strong></td>
</tr>
<tr>
<td>11:00 – 11:30</td>
<td><strong>Options for Accelerating MATLAB Simulations</strong></td>
</tr>
<tr>
<td>11:30 – 12:15</td>
<td><strong>Electronic Interference and Array Signal Processing</strong></td>
</tr>
<tr>
<td>12:15 – 13:15</td>
<td><strong>Lunch</strong></td>
</tr>
<tr>
<td>13:15 – 14:15</td>
<td><strong>End-to-End Modeling of Radar Systems</strong>&lt;br&gt;• Radar Analysis with User-Interface Apps&lt;br&gt;• Integrating the RF Subsystem</td>
</tr>
<tr>
<td>14:15 – 14:30</td>
<td><strong>Break</strong></td>
</tr>
<tr>
<td>14:30 – 16:30</td>
<td><strong>Hands-on Workshop: Phased Array Systems</strong></td>
</tr>
</tbody>
</table>
End-to-End Modeling of Radar Systems

13:15 – 14:15
Radar Block Diagram

Waveform Generator → Transmit Amplifier

Rx Power = Tx Power - 10*\log_{10}(\text{AircraftDistance}^4) + 10*\log_{10}(\text{AircraftCrossSection}*(\lambda^2)/(4*\pi)^3)

RX Signal Processor → RF Front End → Receive Array

Radar Target

Link Budget:
- Input Pulse: -30 dBW
- Tx Gain: +32.3 dB
- Tx Antenna Gain: +35 dB
- Power TX Signal: +37.3 dB
- Path Loss: -148 dB
- Target Return: -49.9 dB
- Power RX signal: -160.6 dB
- Rx Antenna Gain: +35 dB
- Rx kTB: -136.2 dB
- Rx NF: 6 dB
- SNR: 11.3 dBW
A Multi-Disciplined Challenge

Digital

Waveform Generator

Analog / RF

Transmit Amplifier

Mathematical

Radar Target

RX Signal Processor

RF Front End

Receive Array
Radar Analysis with User-Interface Apps

Phased Array System Toolbox Apps

- Radar Waveform Analyzer
  - Rectangular FM – Modulated Pulse Coded
- Radar Equation Calculator
  - Range
  - Peak Transmit Power
  - SNR

Waveform Generator
Transmit Amplifier
Simulating Radar Systems with MATLAB and Simulink

Demo

>>Simulink Radar Model
Radar Component Specification

Redesign system to meet changing requirements

- Increase detection range from 5 km to 10 km
- Maintain 50 m range resolution
- Reuse existing RF hardware (i.e., retain peak transmit power)

Demo (Apps)
>>Simulink Radar Model
Radar Array Processing Algorithms

- **Beamforming**
  - phased.PhaseShiftBeamformer

- **Direction Of Arrival Estimation**
  - phased.BeamscanEstimator

- **Constant False Alarm Rate (CFAR)**
  - phased.CFARDetector

- **Space-Time Adaptive Processing**
  - phased.STAPSMIBeamformer
Radar Targets and Environment

- Radar Targets
  - phased.RadarTarget

- Jammers
  - phased.BarrageJammer

- Clutter
  - phased.gpu.ConstantGammaClutter
Modeling the RF Subsystem
How do we bridge the gap between engineers who speak different languages?
How do we bridge the gap between engineers who speak different languages?
Solution: Multi-domain Modeling

End-to-end model serves as:
- Executable Specification
- Test Environment

“Here are some initial RF specs”

“Here’s the RF model”
RF Subsystem Design using SimRF

- SimRF allows us to model RF architectures and RF components using:
  - Standard RF parameters (e.g., $IP_3$, noise figure)
  - Measured component data (e.g., measured $s$-parameters)
SimRF Simulation Technologies / Libraries

Equivalent Baseband (faster)

Circuit Envelope (more accurate)

Demo (RF) >> Simulink Radar Model
# Workflow with MATLAB and Simulink

<table>
<thead>
<tr>
<th>Engineering Tasks</th>
<th>Features of MATLAB and Simulink</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Workflow with MATLAB and Simulink

- Specifications and System Tradeoffs
## Workflow with MATLAB and Simulink

### Engineering Tasks
- Specifications and System Tradeoffs
- Explore Digital Algorithms and RF Architectures

### Signal Processing & SimRF Libraries

- **Specifications**
- **System Tradeoffs**
- **Explore Digital Algorithms and RF Architectures**

![RF Front End](image-url)
Workflow with MATLAB and Simulink

<table>
<thead>
<tr>
<th>Engineering Tasks</th>
<th>Simulink plus SimRF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specifications and System Tradeoffs</td>
<td></td>
</tr>
<tr>
<td>Explore Digital Algorithms and RF Architectures</td>
<td></td>
</tr>
<tr>
<td>Multi-domain system simulation combining digital and analog subsystems</td>
<td></td>
</tr>
</tbody>
</table>
Summary - What We Used Today

- Simulink & MATLAB
- Communications System Toolbox
- LTE System Toolbox
- MATLAB Coder
- Parallel Computing Toolbox
- Phased Array System Toolbox
- SimRF
Key Take-Aways

MATLAB® and Simulink® provide:

- Single platform that facilitates collaboration between multiple engineering teams
  - System engineers
  - Signal processing engineers
  - RF engineers

- Robust and user-friendly environment to explore design tradeoffs between multiple domains
  - Digital: Communications and Radar algorithms
  - Analog: RF receiver architecture
Thank You for Attending!
## Agenda

<table>
<thead>
<tr>
<th>Time</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00 – 10:00</td>
<td><strong>Building Communication System Simulations</strong></td>
</tr>
<tr>
<td></td>
<td>• Constructing a basic QPSK communications model in Simulink</td>
</tr>
<tr>
<td></td>
<td>• 802.11a Wireless LAN example</td>
</tr>
<tr>
<td></td>
<td>• Adding RF design detail</td>
</tr>
<tr>
<td>10:00 – 10:45</td>
<td><strong>4G LTE Design using the LTE System Toolbox</strong></td>
</tr>
<tr>
<td>10:45 – 11:00</td>
<td>Break</td>
</tr>
<tr>
<td>11:00 – 11:30</td>
<td><strong>Options for Accelerating MATLAB Simulations</strong></td>
</tr>
<tr>
<td>11:30 – 12:15</td>
<td><strong>Electronic Interference and Array Signal Processing</strong></td>
</tr>
<tr>
<td>12:15 – 13:15</td>
<td>Lunch</td>
</tr>
<tr>
<td>13:15 – 14:15</td>
<td><strong>End-to-End Modeling of Radar Systems</strong></td>
</tr>
<tr>
<td></td>
<td>• Radar Analysis with User-Interface Apps</td>
</tr>
<tr>
<td></td>
<td>• Integrating the RF Subsystem</td>
</tr>
<tr>
<td>14:15 – 14:30</td>
<td>Break</td>
</tr>
<tr>
<td>14:30 – 16:30</td>
<td><strong>Hands-on Workshop: Phased Array Systems</strong></td>
</tr>
</tbody>
</table>
Phased Array Systems Workshop

14:30 – 16:30

Armerkom
Ankara, Turkey
February 28, 2014

Nick M. Zayed
The Pre-requisites

- Basic familiarity of the MATLAB language and environment

- Attended seminars/webinars/presentations on the Phased Array System Toolbox

- Have engaged in the phased array related projects/programs, such as radar system design

- Eagerness to learn and use the product
Agenda

- Introduction *(5 minutes)*

- Exercise # 1: Array Design and Analysis *(40 minutes)*

- Exercise # 2: Digital Beamforming *(40 minutes)*

- Exercise # 3: Modeling a Mono-Static Radar System *(20 minutes)*

- Q & A *(15 minutes)*
# Phased Array System Toolbox

*A rich set of features with an open and extensible framework*

## Phased array design and analysis
- Linear, rectangular, conformal geometries
- Shading, tapering
- Element position and orientation
- Gain, delay, steering vector
- Subarrays

## Temporal processing
- Time varying gain, pulse compression
- Coherent, non-coherent integration
- Signal detection and ROC curves
- CFAR processing, range/Doppler estimation

## Waveform design and analysis
- Pulsed CW, FMCW
- LFM and stepped FM
- Phase coded
- Staggered PRF
- Ambiguity function
- Matched filter

## Spatial processing
- Digital beamforming: narrowband & broadband, Conventional, MVDR, LCMV, Frost, time delay, time delay LCMV, subband phase shift
- DOA processing: Monopulse, MVDR, beamscan, ESPRIT, Root-MUSIC

## Signal modeling framework
- Monostatic and multistatic scenarios
- Transmitter and receiver preamp models
- Point target and Swerling target models
- Narrowband and broadband modeling
- Platform motion
- Clutter and jammer models

## Space-time adaptive processing
- Displaced phase center array (DPCA)
- Adaptive DPCA
- Sample matrix inversion (SMI)
- Angle-Doppler response
Exercises #1: Array Design and Analysis

a. Creating/defining a 10 element Uniform Linear Array

b. Analyzing and visualizing the array response

c. Examining the use of steering vector on an ULA

d. (Optional) observing the ULA responses for azimuth cut with different elevations

e. (Optional) creating a conformal array of 8 elements (a circular array)
Exercises #1: Array design and Analysis

- Screenshots
Exercises #2: Conventional and Adaptive Beamformers

a. Simulating the Radar return Signal

b. Create a beamforming array

c. Perform beamforming with a Phase Shift Beamformer

d. Modeling the Interference Signals

e. Use MVDR Beamformer when interferers are present

f. (Optional) Apply and observe LCMV beamformer
Exercises #2: Conventional and Adaptive Beamformers

- Screenshots
Exercises #3: Modeling Mono-static Radar System

- Waveform design/generation
- Receiver design
  - Thermal noise, no clutter
- Transmitter design
  - Determine required SNR and transmit peak power
  - Design Radiator and Collector
- Targets and environment
- Signal processing and detection
  - Perform beamforming & matched filtering
  - Perform range normalization
  - Range detection & Doppler estimation
Exercises #3: Modeling Mono-static Radar System

- Screenshots
Thank You for Attending!