MATLAB EXPO 2016
Modelling and Simulating RF Sensor Systems

Marc Willerton
Overview

- Challenges developing RF Sensor Systems
- Analysing RF data streams
- Designing RF components and algorithms
- Simulation of RF Systems
RF Sensor Systems

Wireless Broadband

Radar Systems

Satellite Communications

Mobile Handsets/Basestations

Advanced Driver Assistance Systems

Antenna Arrays

MATLAB EXPO 2016
Challenges with Developing RF Sensor Systems

Antenna, Antenna arrays
Non-isotropic pattern, coupling, edge effects
Array Perturbations

Channel
interference, clutter, noise

Mixed-Signal
Continuous & discrete time, jitter, finite precision

RF Electronics
frequency dependency, non-linearity, noise, mismatches, synchronization of multiple LO’s

Algorithms
Finite precision loss, latency, resource utilisation

Receiver

Transmitter

Data/Waveform
System Simulation of an RF System

Challenge 1: How can I test the effect of all of these RF impairments on my system performance?
Challenges with Developing RF Sensor Systems

Antenna, Antenna arrays
Non-isotropic pattern, coupling, edge effects
Array Perturbations

Channel
interference, clutter, noise

Mixed-Signal
Continuous & discrete time, jitter, finite precision

RF Electronics
frequency dependency, non-linearity, noise, mismatches, synchronization of multiple LO’s

Algorithms
Finite precision loss, latency, resource utilisation

Waveforms
Challenges with Developing RF Sensor Systems

- **Antenna, Antenna arrays**
  - Non-isotropic pattern, coupling, edge effects
- **Array Perturbations**
- **Channel**
  - Interference, clutter, noise
- **Mixed-Signal**
  - Continuous & discrete time, jitter, finite precision
- **RF Electronics**
  - Frequency dependency, non-linearity, noise, mismatches, synchronization of multiple LO’s
- **Algorithms**
  - Finite precision loss, latency, resource utilisation

MATLAB EXPO 2016
Detecting Signal Interference in MATLAB

LTE Base Station
- 5MHZ Transmit Bandwidth
- 20dBm Transmit Power

Narrowband Interferer
- Not persistent

Challenge 2: When and where does the interferer arise in my system?
```matlab
close all

% OPEN A HANDLE TO THE DATA FILE
Fs = 10240000;
filename = 'C:\Users\John\Documents\SignalProcessing\Data\frequency.bin';
fid = fopen(filename, 'r');

% DEFINE THE SPEAKER ANALYZER
speakerAnalyzer = dsp.SpectrumAnalyzer('SampleRate', Fs, 'SpectrumType', 'Power Spectra', 'PowerUnits', 'dBm');

for i = 1:1000
    % WHILE ISDONE(filename) % LOAD IN AND VISUALIZE THE DATA FRAME-BY-FRAME
    while isDone(filename)
        % LOAD IN AND VISUALIZE THE DATA FRAME-BY-FRAME
        data_frame = fread(fid, 1, 'float32'); % Load data frame
        step(speakerAnalyzer, data_frame); % Visualize data frame
        pause(0.1); % Pause for 0.1 seconds
    end
    % REDRAW(False)
end
```
Clear all.

% Open a handle to the data file
Fs = 183800200;
filename=app.BinFileReader('C:\\Users\\rare\\Desktop\\SampleData.bin', 'SamplePerFrame', 3572, 'TDataComplex', 'true');
% Define the spectrum analyzer
spectrumscope = spectrumanalyzer('SampleRate', Fs, 'SpectrumType', 'Power density', 'PowerUnits', 'dBW');

% Set spectrum scope.
spectrumscope.SpectrumMask Source = 'Upper';
upperMask = [Fs/2, -110; 3.5e+6, -110; 3.5e+6, -85; 2.5e+6, -85; 2.5e+6, -110; Fs/2, -110];
set(spectrumscope, 'SpectrumMask', upperMask);

% Release spectrum scope.
release(spectrumscope);

% Define a mask according to some specification
spectral_mask = [100 5000];
% Select spectral mask that falls to meet specification
record_frame = 11; % Record frame number which fail to meet specification
frame_number = 1;

PLOTFRAME = 8; % Plot spectrum at this frame number.

for i = 1:1000
    while isDone(filename) % Load in the data frame by frame
        data = fopen(filename);
        stepspectrumscope.data_step;...
        % Compare frame against mask. If fails then record frame number.
        black_data = 10*log10(pwelch(data_step, [1, 1], Fs, 'nfft'));
        if ~black_data < spectral_mask
            record_frame = [record_frame, frame_number];
        end
        % Plot the Welch PSD of the frame if PLOTFRAME
        if frame_number == PLOTFRAME
            [black_data, freq_bins] = pwelch(data_step, 1024, 0.1024, Fs, 'nfft');
            figure
            plot(freq_bins, 10*log10(black_data, freq_bins, spectral_mask));
        end
        advance frame number
        frame_number = frame_number + 1;
    end
end
Challenges with Developing RF Sensor Systems

- **Antenna, Antenna arrays**
  - Non-isotropic pattern, coupling, edge effects
- **Array Perturbations**
- **Channel**
  - Interference, clutter, noise
- **Mixed-Signal**
  - Continuous & discrete time, jitter, finite precision
- **RF Electronics**
  - Frequency dependency, non-linearity, noise, mismatches, synchronization of multiple LO’s
- **Algorithms**
  - Finite precision loss, latency, resource utilisation

Waveforms
Rejecting Interference using Antenna Arrays

N Antennas

I/Q Antenna Samples

Detection (e.g. AIC/MDL)

Estimation (e.g. MuSiC)

Estimate the number of sources

Estimate unknown parameters of interest (e.g. DOA)

Reception (e.g. MVDR Beamforming)

P_1, P_2, ..., P_M
θ_1, θ_2, ..., θ_M
φ_1, φ_2, ..., φ_M

Receive the desired signal from a given direction in space

Challenge 3: How well can my array cancel out my interference?
Define Scenario Setup for Array and Sources

1. \( fc = 2 \times 9 \% \) Frequency
2. \( \lambda = 3 \times B / fc, \% \text{Wavelength} \)
3. \( N = 6 \% \) Number of array elements
4. \( L = 1000 \% \) Number of snapshots
5. \( array = \text{phased} \_\text{ULA}(\text{NumElements}, \text{ElementSpacing}, \lambda / 2); \)
6. \( \text{DOA} = [20 40 60, 0 0 0]; \% \text{Sources at 20, 40 and 60 degrees Azimuth and 0 degrees elevation} \)
7. \( \text{viewArray}(array) \)

Generate received signals and construct covariance matrix

8. \( x = \text{sensorsig}((\text{getElementPosition}(array) / \lambda, \text{DOA}, \text{db2pow}(-10)))'; \)
9. \( R = x \times x' / L \)

Detection Algorithm (How many signal sources)

10. \( \text{nsig} = \text{axtest}(x') \)

Estimation Algorithm (What are the unknown source parameters)

11. \( \text{az}_{\text{range}} = -90:90; \)
12. \( z = \text{music}([\text{getElementPosition}(array) / \lambda, \text{nsig}, -90:90]); \)
13. \( \text{[peakval, peakpos]} = \text{findpeaks}(z, 'NPeaks', \text{nsig}, ['SortStr', 'descend']); \)
14. \( \text{DOA}_{\text{Est}} = \text{sort}(\text{az}_{\text{range}}(\text{peakpos})); \)
15. \( \text{plotmusic}(z, \text{DOA}) \)

Reception Algorithm (Receive the signals)

16. \( w1 = \text{superresolution}\_\text{beamformer}((\text{getElementPosition}(array) / \lambda, \text{DOA}_{\text{Est}}(1), \text{DOA}_{\text{Est}}(2:\text{end}))); \)
17. \( \text{plotbeamformer}(array, fc, w1, \text{DOA}) \)
Sensitivity Analysis of Array Systems

N Antennas

I/Q Antenna Samples

Array Imperfections (e.g. sensor location, gain and phase)

Detection (e.g. AIC/MDL)

Estimation (e.g. MuSiC)

Reception (e.g. MVDR Beamforming)

M Sources

P_1, P_2, ..., P_M

θ_1, θ_2, ..., θ_M

φ_1, φ_2, ..., φ_M

Estimate the number of sources

Estimate unknown parameters of interest (e.g. DOA)

Receive the desired signal from a given direction in space

Challenge 4: What effect do array imperfections have on array performance?
Define Scenario Setup for Array and Sources

\[
\begin{align*}
\text{fc} &= 2e9; \quad \text{Frequency} \\
\text{lambda} &= \text{fc}/\text{fc}; \quad \text{Wavelength} \\
N &= \text{Number of array elements} \\
L &= 1000; \quad \text{Number of snapshots} \\
\text{harray} &= \text{phased.ULA('NElements',N,'ElementSpacing',\text{lambda}/2)}; \\
\text{DOA} &= [20 40 60, 0 0 0]; \quad \text{Sources at 20, 40 and 60 degrees Azimuth and 0 degrees elevation} \\
\text{rng}(1001)
\end{align*}
\]

Generate and Visualise Perturbations

\[
\begin{align*}
\text{sensorPert} &= \text{zeros}(3,N); \\
\text{sensorPert} &= [2,2:end] = 0.2*rand(2,N-1); \\
\text{sensorPos} &= \text{getElementPosition(harray)/lambda} - \text{sensorPert}; \\
\text{plot}(
\text{sensorPos(1,:)','lambda, sensorPos(2,:), 'lambda, 'MarkerSize', 15)
\end{align*}
\]

Generate received signals and construct covariance matrix

\[
\begin{align*}
x &= \text{sensorSig(sensorPos,L,DOA,db2pow(-20))};' \\
R &= x^*x/L
\end{align*}
\]

Detection Algorithm (How many signal sources)

\[
\text{nsig} = \text{aicest}(x').
\]

Estimation Algorithm (What are the unknown source parameters)

\[
\begin{align*}
\text{az_range} &= -90:90; \\
\theta &= \text{music}(R,\text{getElementPosition(harray)/lambda, nsig,-90:90});
\end{align*}
\]
Challenges 5: What is the radiation pattern of this antenna? Challenge 6: What is the effect of putting this antenna in an array?
Investigating Antenna Patterns, Coupling and Edge Effects

Independent Elements

6 Element ULA

Non-independent Elements
Challenges with Developing RF Sensor Systems

Antenna, Antenna arrays
Non-isotropic pattern, coupling, edge effects
Array Perturbations

Channel
interference, clutter, noise

Mixed-Signal
Continuous & discrete time, jitter, finite precision

RF Electronics
frequency dependency, non-linearity, noise, mismatches, synchronization of multiple LO’s

Algorithms
Finite precision loss, latency, resource utilisation

Waveforms
Measuring the effect of non-linearity within an RF Frontend

Challenge 7: What effect do RF impairments have on out of band leakage?

Phase Noise
Non-linearity
Carrier Leakage
Impedance Mismatches
Measuring the effect of non-linearity within an RF Frontend

Challenge 7: What effect do RF impairments have on out of band leakage?
model = 'PA_Ideal';
open_system(model);

% Initialization LTE testbench + RF receiver component
%
% Configuration (T36.101 RNC B.6, SMS 64-QAM rate 3/4)

rmc = lteRMCDL('B.6');

% Create eNodeB transmission with fixed PDSCH data

rs=20; % fixed random seed (arbitrary)
data = randi([0 1], sum(rmc.PDSCH.TransRlkSizes),1);

[tx, info] = lteRMCDLtool(rmc, data);

ExtraSamples = 400;
rmc.BFrame = 1;
NextFrame = lteRMCDLtool(rmc, data);

tx = [tx; NextFrame(1:ExtraSamples)];

FrameLength = length(tx);

% Write the sampling rate and chip rate to the configuration structure to
% allow the calculation of ACLR parameters

rmc.SamplingRate = info.SamplingRate;
rmc.UTRAChirpRate = 3.88;

SampleTime = 1/info.SamplingRate;

% Calculate ACLR measurement parameters

[aclr, nRC, E_C, EutraA] = ACLRParameters(rmc);

% Apply required oversampling

resampled = UpSamplingFilter(tx);

FrameLength = FrameLength*aclr.OSR;
SampleTime = SampleTime/aclr.OSR;

% Simulate LTE frame

% Generate test data for RF RX

time = (4*FrameLength/ExtraSamples)*SampleTime;
System Simulation of an RF System

Challenge 1: How can I test the effect of all of these RF impairments on my system performance?
Targeting Signal Processing Algorithms to SDR Platforms

- Execute fixed radio functions on FPGA
- Tunable pre-defined radio parameters
- Easy out-of-the-box experience

1. Streaming Mode
Targeting Signal Processing Algorithms to SDR Platforms

- Generate code to implement custom functionality on FPGA
- Customized using HDL Coder
Targeting Signal Processing Algorithms to SDR Platforms

- Generate code to implement custom functionality on FPGA and ARM
- Customized using HDL Coder and Embedded Coder
- Generate AXI Interface between hardware and software
**Challenge**
Implement FPGA based radio signal processing in a small team mainly consisting of people with signal processing and programming background.

**Solution**
Use HDL Coder to generate VHDL for signal processing.

**Results**
- Successful implementation running on FPGA
- Generated code easy to integrate into main design
- Very short lead time for changes in design


Conclusions

- We rely increasingly on more complex sensor systems in our everyday lives
- Engineers developing these sensor systems must overcome many challenges to ensure the system will reach its desired performance
- Simulation of these systems at the appropriate level of fidelity can help detect design issues early
Questions?

Email: marc.willerton@mathworks.co.uk
Challenges with Developing RF Sensor Systems

Antenna, Antenna arrays
type of element, # elements, coupling, edge effects
- Antenna Toolbox
- Phased Array System Toolbox

Channel
interference, clutter, noise
- Communications System Toolbox
- Phased Array System Toolbox

RF Electronics
frequency dependency, non-linearity, noise, mismatches
- SimRF
- RF Toolbox

Mixed-Signal
Continuous & discrete time
- Simulink (Simscape)
- DSP System Toolbox
- Control System Toolbox

Algorithms
beamforming, beamsteering, MIMO
- Phased Array System Toolbox
- Communications System Toolbox
- LTE System Toolbox
- WLAN System Toolbox

Waveforms
- Phased Array System Toolbox
- Instrument Control Toolbox
- LTE System Toolbox
- WLAN System Toolbox