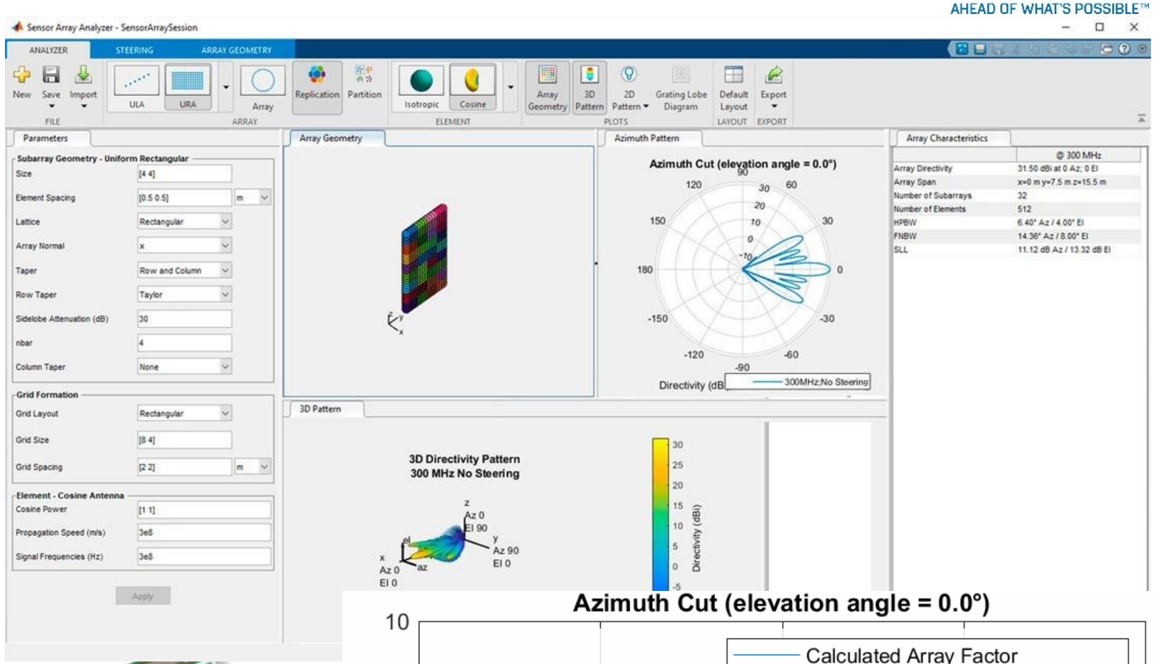
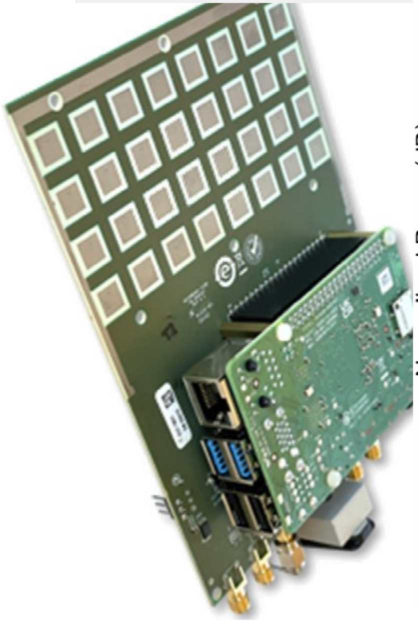
General-Purpose Phased Array Learning Kit: Efficient Interference Mitigation

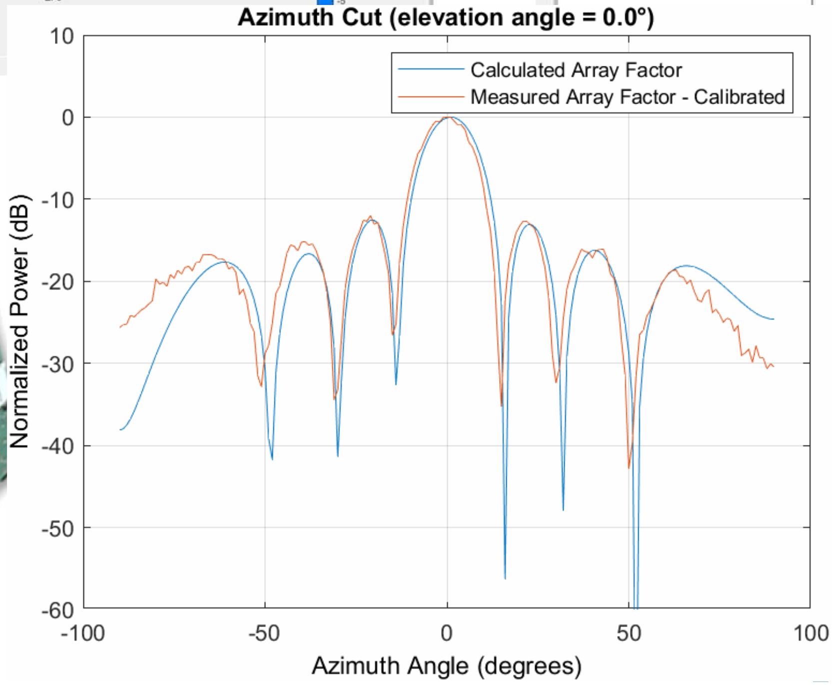
Robin Getz, MathWorks rgetz@mathworks.com
Jon Kraft, Analog Devices jon.kraft@analog.com

Presentation Goals

1. Gain an understanding of phased array and beamforming concepts
2. Learn how simulation models can be used to predict array and beam behavior for system design and test
3. Validate simulation models using prototype hardware
4. Learn about practical applications for phased array systems







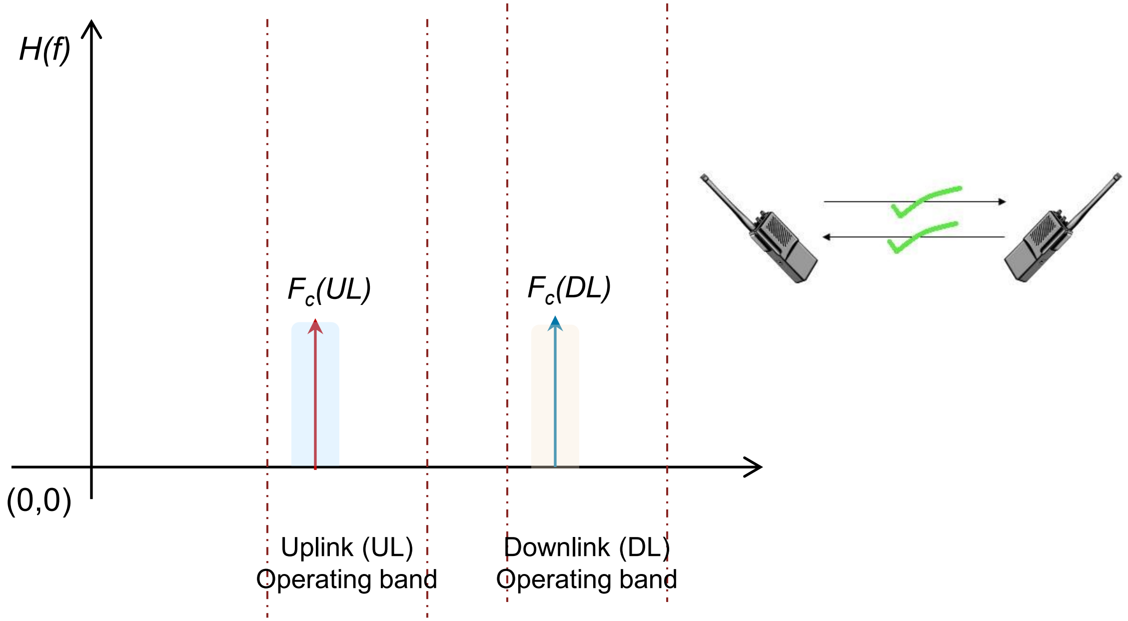
Agenda

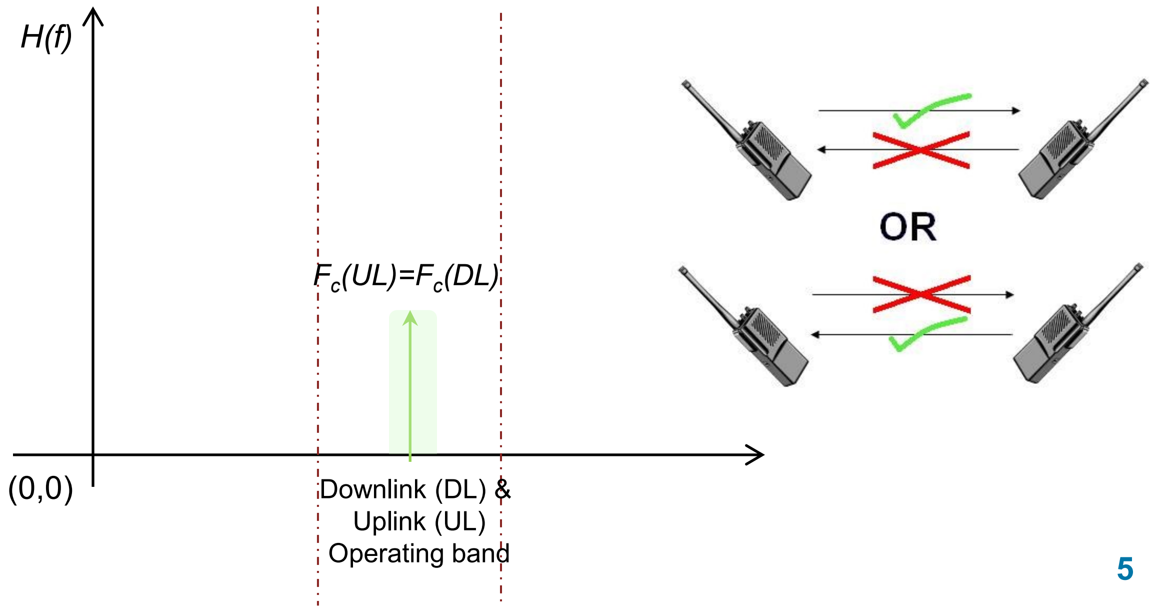
- Phased Array Systems Overview
  - Basics of Phased Arrays
  - Antenna arrays and models
  - Calibration
- Impairments
  - Beam Tapering
  - Grating Lobes
- Application Examples
  - Null Steering
- Summary and Topics for Further Study
- Q&A

Phased Array Systems Overview

Sharing Spectrum

- FDD: Frequency Division Duplex
  - frequency bands are paired
  - simultaneous transmission on two frequencies (one for downlink and the other for uplink)
- TDD: Time Division Duplex
  - frequency bands are unpaired
  - uplink and downlink transmissions share the same channel an carrier frequency
  - The transmissions in uplink and downlink directions are time-multiplexed

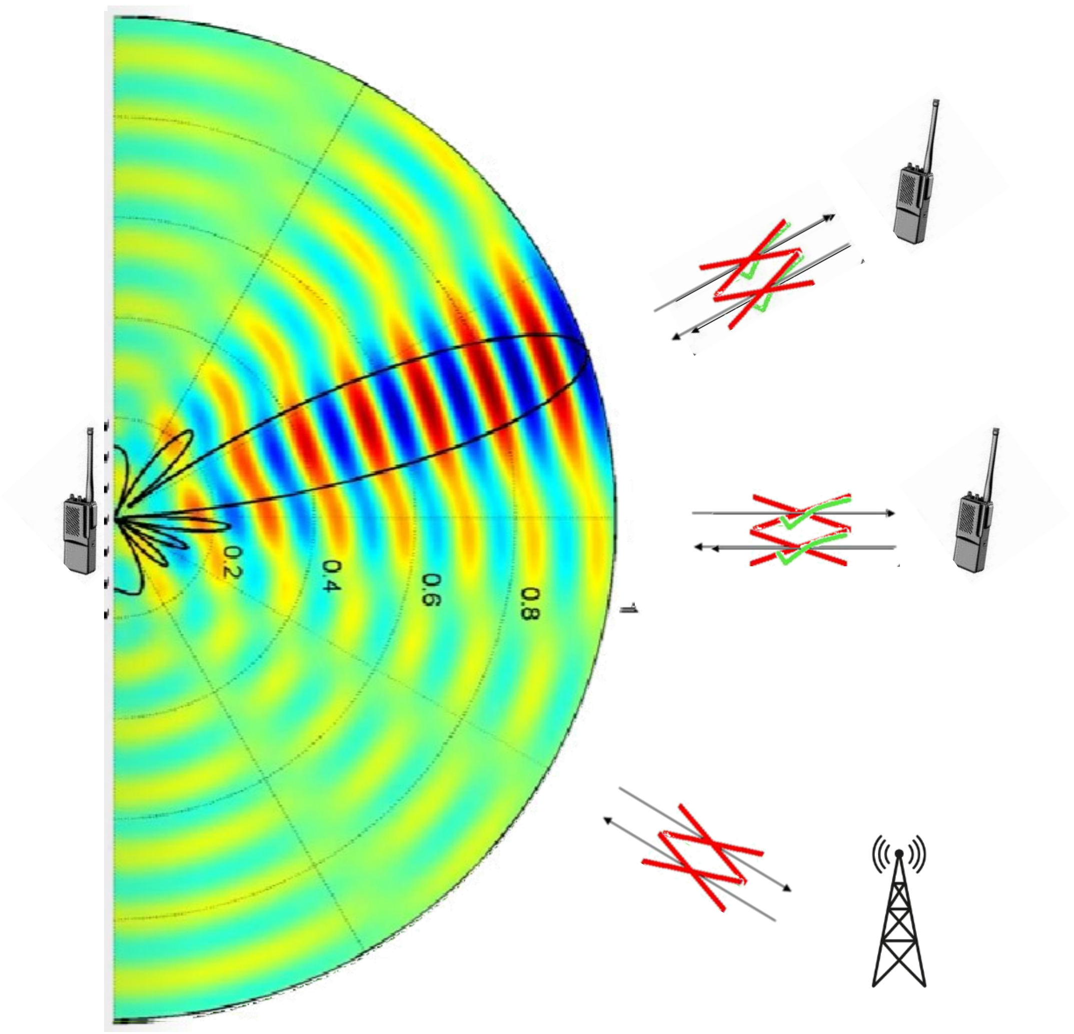




Sharing Spectrum

- Spatial separation :
  - arrays of transmit/receive antennas are employed to transmit or receive a signal towards a certain direction in space through beamforming techniques

- Combine Time, Frequency and Space for max spectrum efficiency



What is Phased Array Beamforming?

- The ability to “steer” multiple antennas without mechanical movement
  - Moving mass around is relatively slow and mechanical systems need maintenance
  - Electronic control allows movement of beams in a fraction of a second
  - Steer beams and nulls

- Using multiple, smaller antennas also allows for multiple, independently controlled beams to be generated

Where are Phased Array Systems Used?

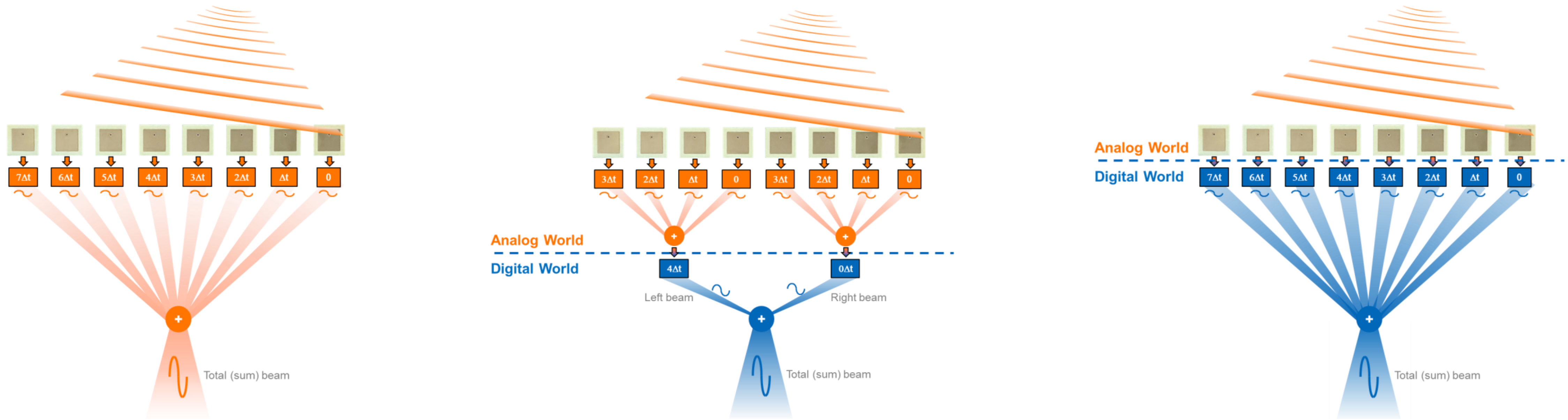
Multifunction Radars

Wireless Communications

Satellite Communications

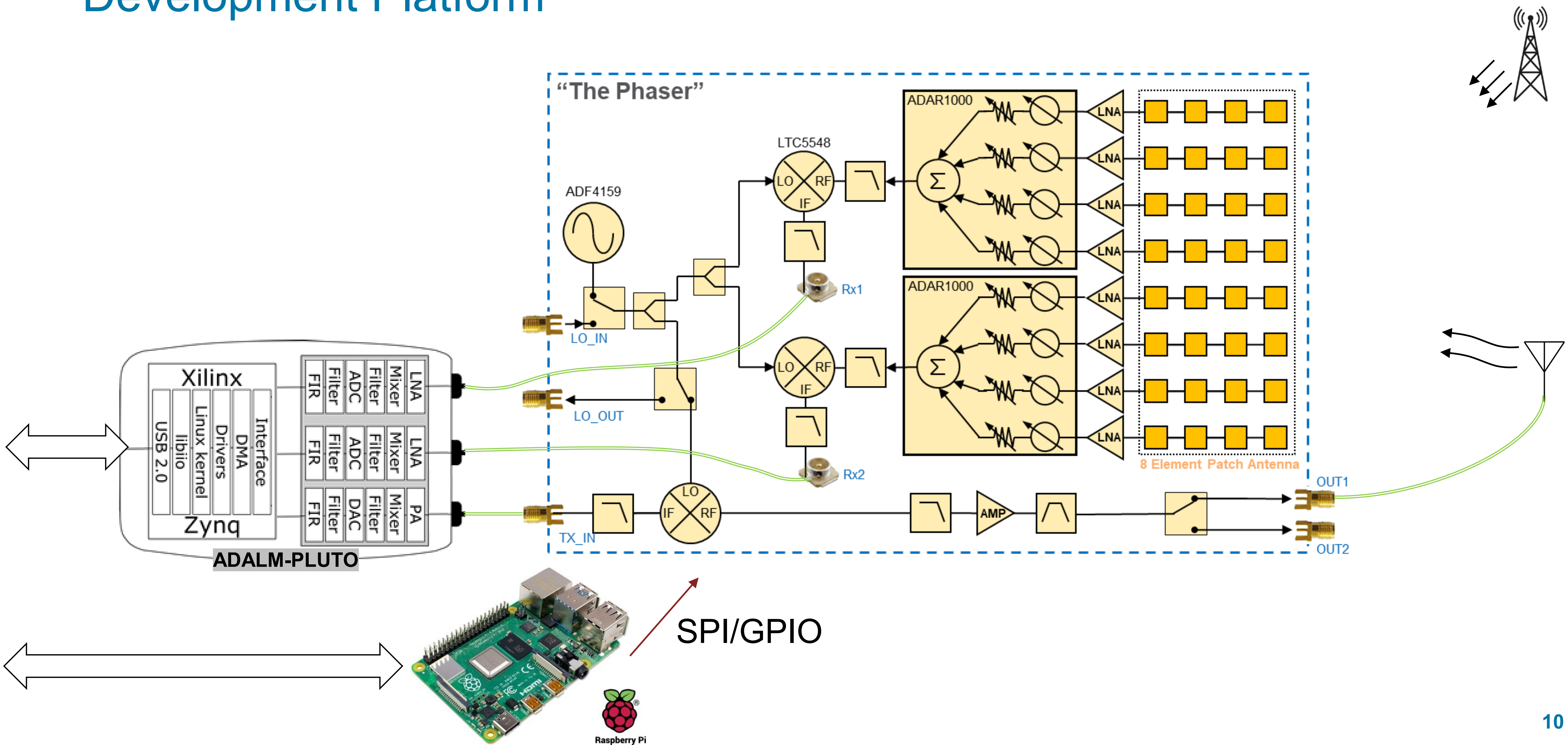
Acoustics

Beamforming Architectures



Analog Beamforming	Hybrid Beamforming	Digital Beamforming
Beam formed by weighting RF paths	Digital combining of multiple analog beams	Beam formed by weighting digital paths
Single set of data converters	1 < m < n sets of data converters	Separate data converters for each element
Low power/complexity	Moderate power/complexity	Highest power / complexity
Good for coverage	Compromise between analog and digital	Highest capacity / flexibility
Single narrow beam	Often the best choice with existing technology	Wide analog beamwidth, narrow digital beams

Hardware Platform – CN0566 Phased Array (Phaser) Development Platform



ADF4159

500MHz

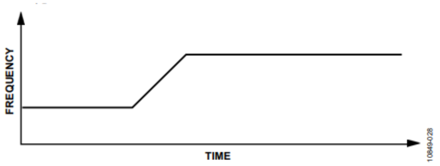


Figure 33. Single Ramp Burst

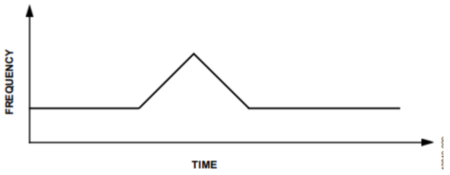


Figure 34. Single Triangular Burst

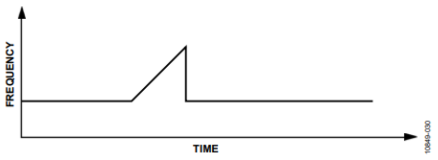


Figure 35. Single Sawtooth Burst

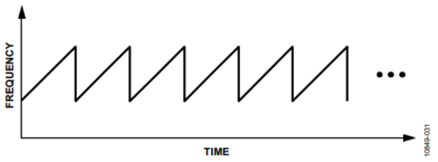


Figure 36. Continuous Sawtooth Ramp

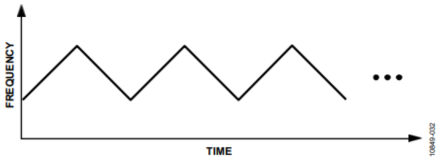


Figure 37. Continuous Triangular Ramp

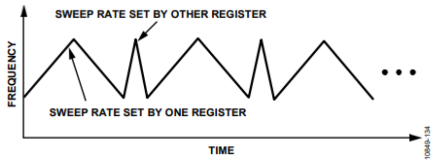
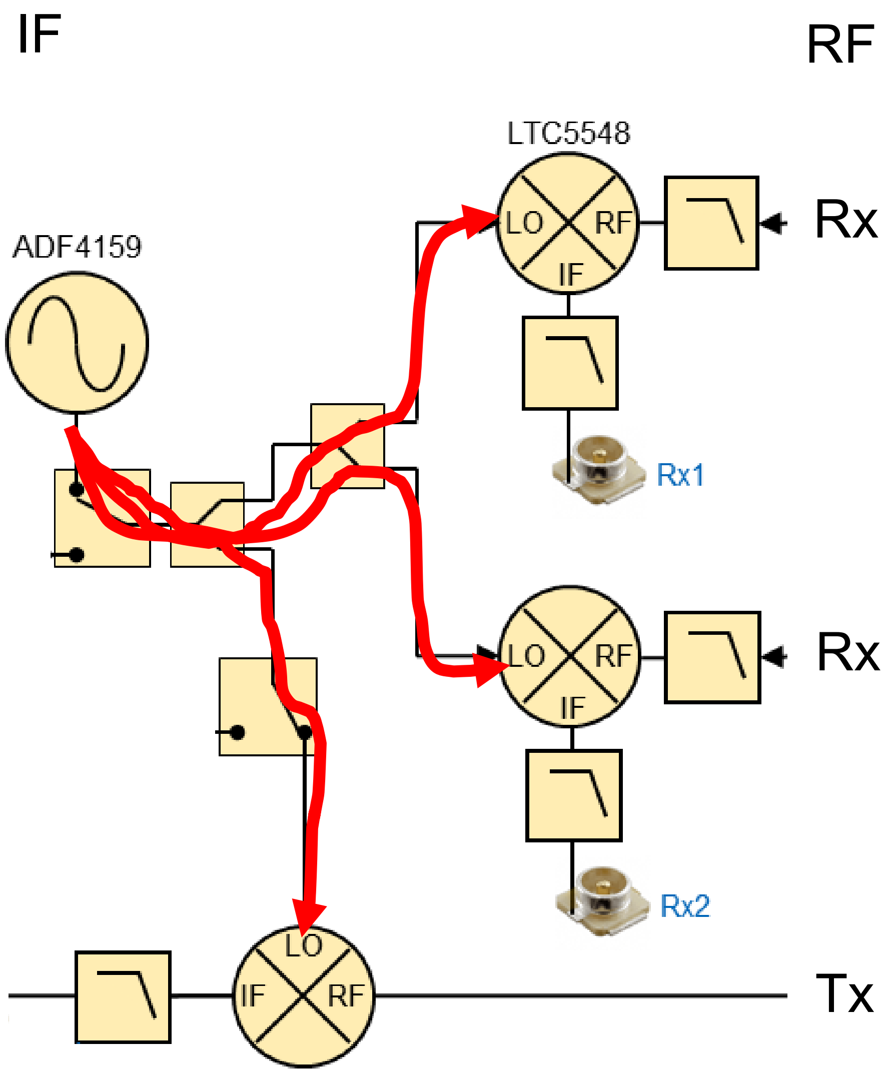


Figure 39. Dual Ramp with Two Sweep Rates



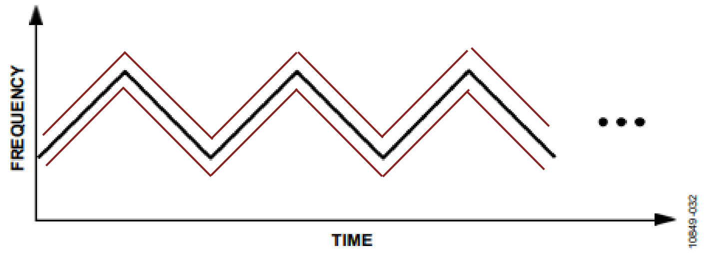
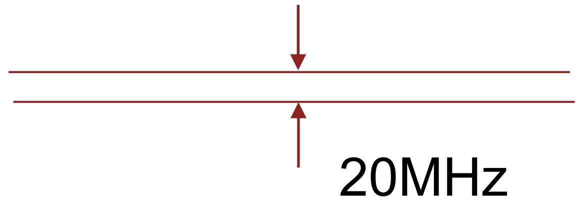
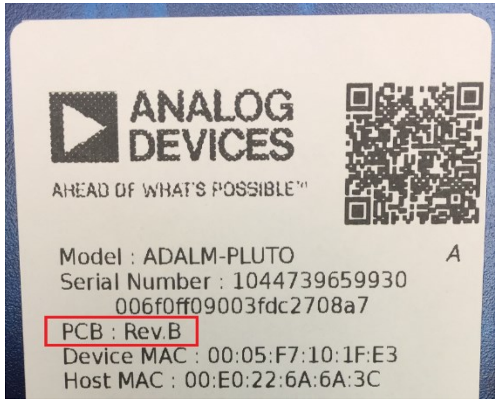


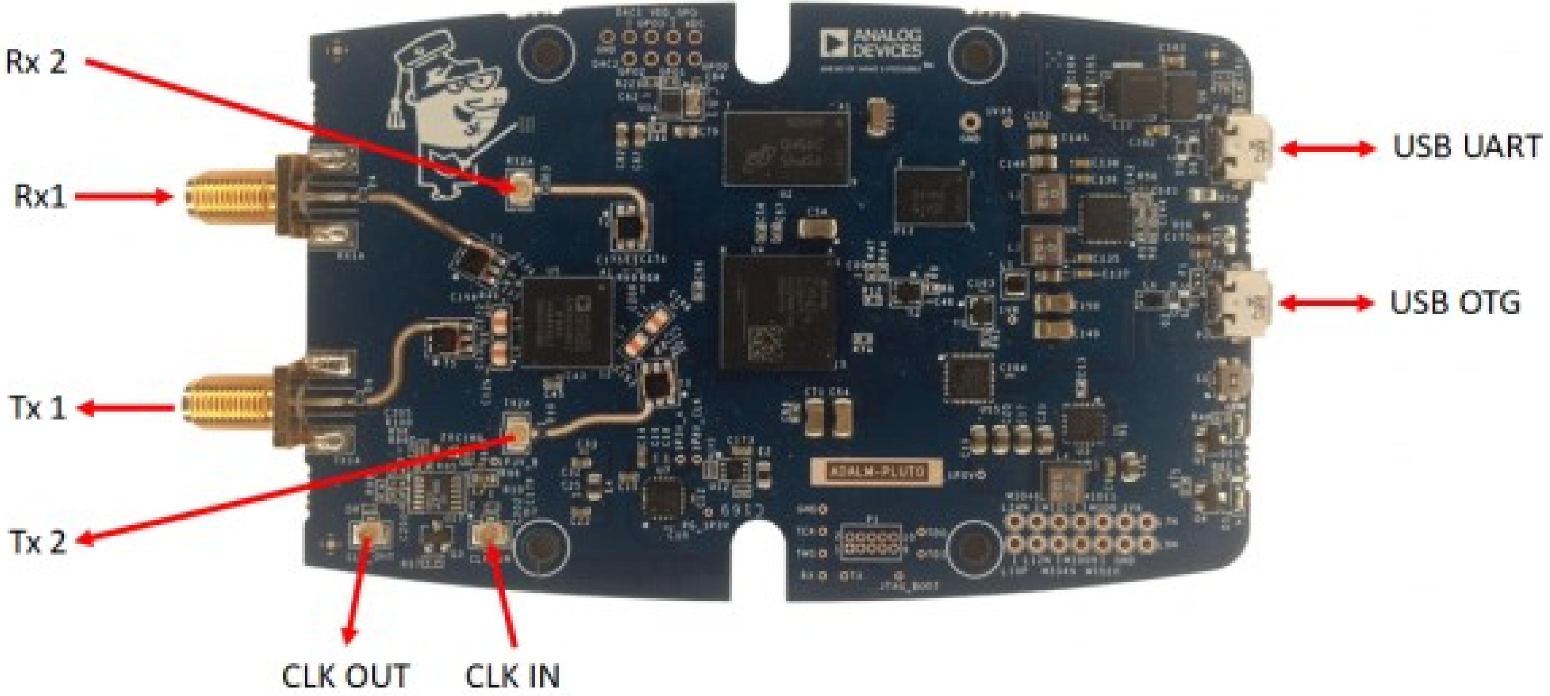
Figure 37. Continuous Triangular Ramp



Pluto Rev C/D – 1 SMA channel, 1 internal u.FL Channel

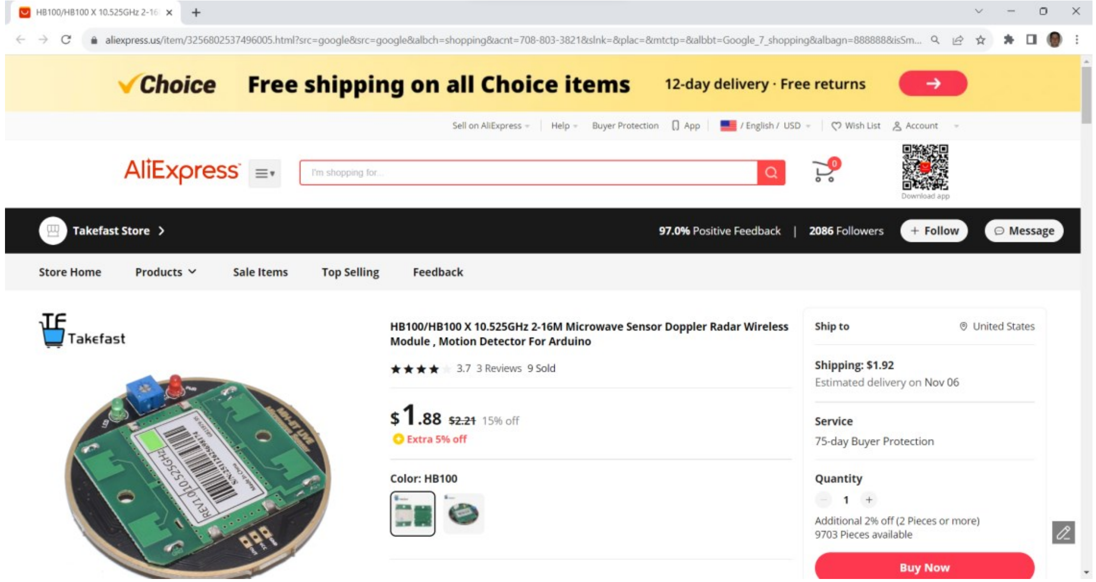


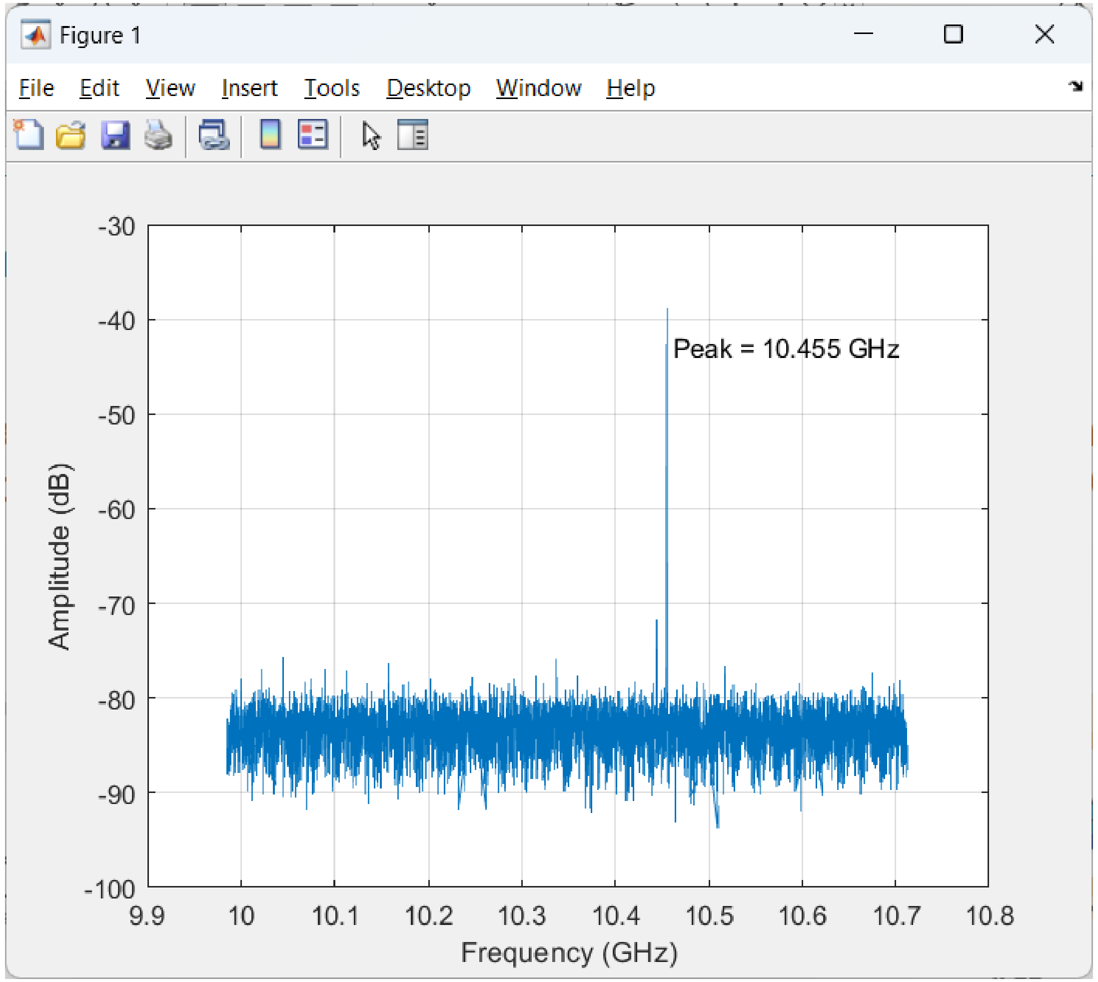
Rev	
A / B	1 channel
C / D	2 channel



https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/ADALM-PLUTO.html

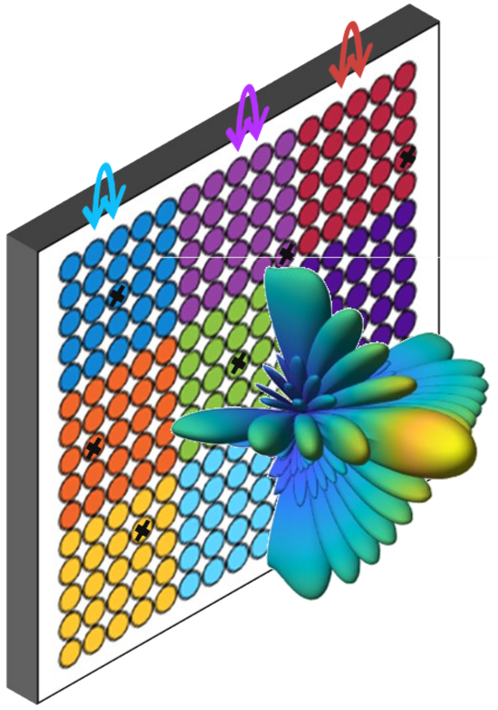
HB100 – 10.525 GHz?

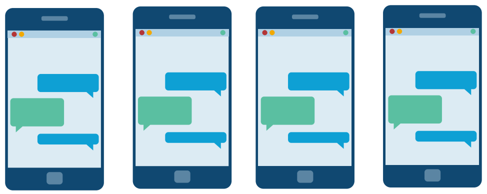


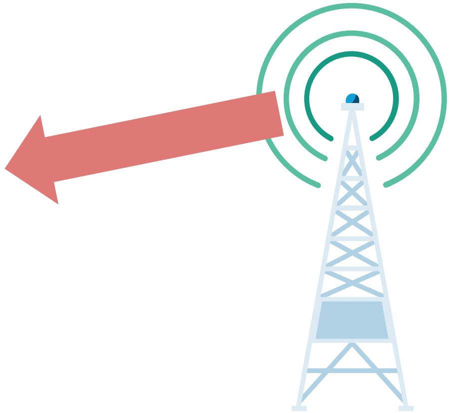


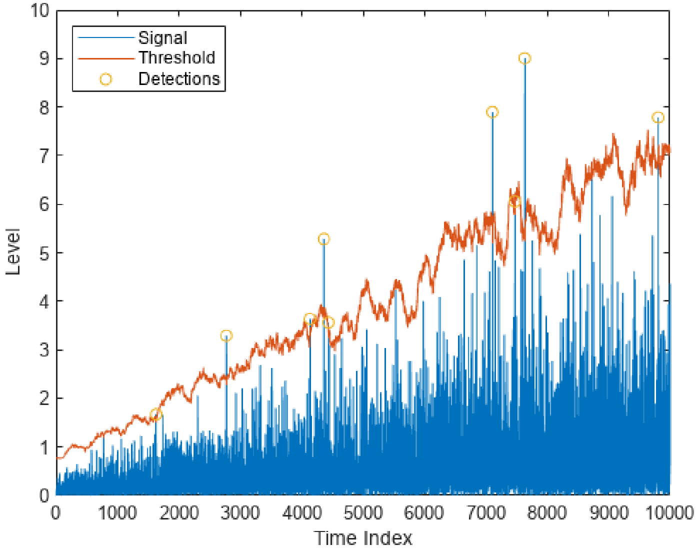
Phased Array Design Challenges

Array Design	Mutual Coupling
Signal Processing	Interference Mitigation
Calibration	Waveform Design









Phased Array Design Skillset Requirements

Array Design	Mutual Coupling
Signal Processing	Interference Mitigation
Calibration	Waveform Design

- RF and Antenna Design
- System Engineering
- Digital Hardware Design
- Signal Processing Algorithm Development
- Software, Firmware, and HDL design

Goal: Establish a Common Design Language and Development Framework

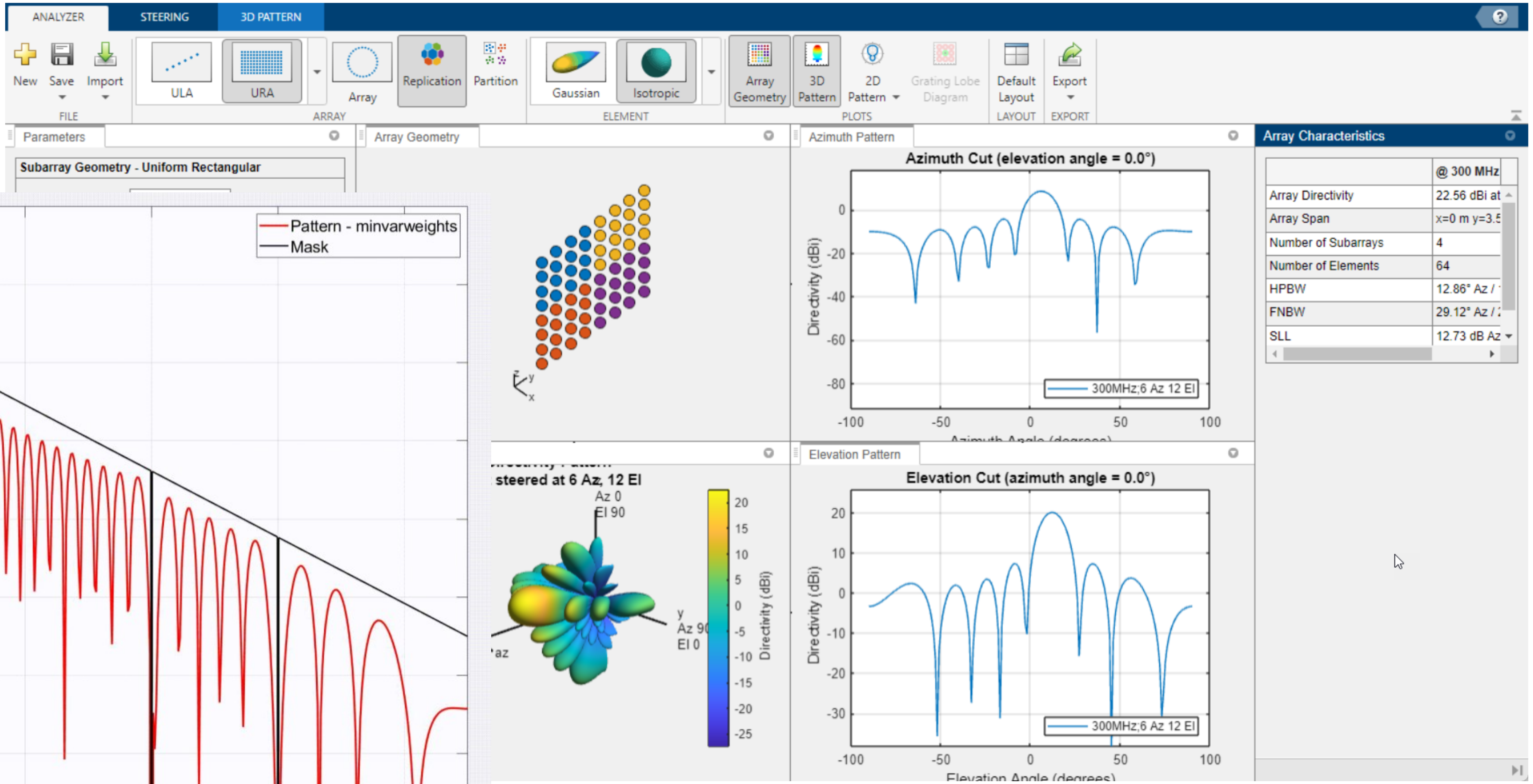
- Create algorithm and device models

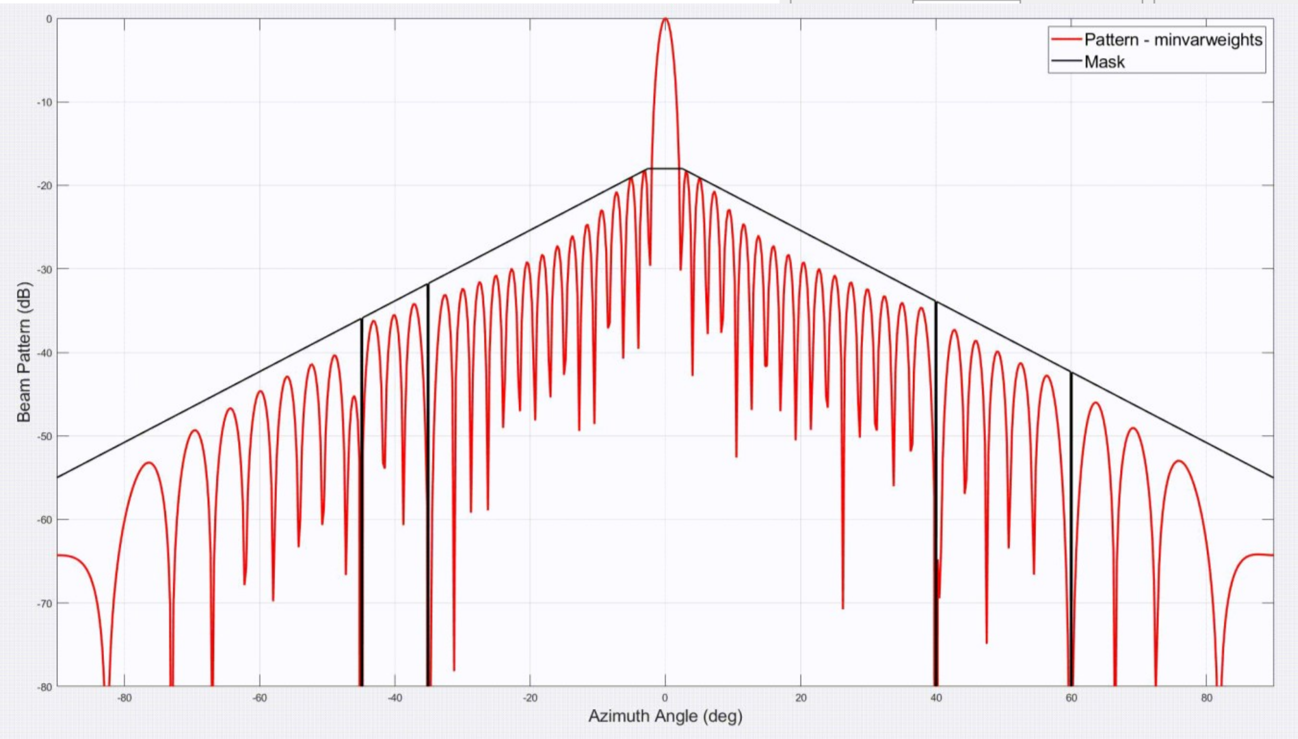
```
%% Array Model and Steering Vector
c = physconst('LightSpeed');
lambda = c/fc_hb100;
phaserModel = phased.ULA('NumElements',8,...
    'ElementSpacing',bf.ElementSpacing);
steeringVec = phased.SteeringVector("SensorArray",phaserModel, ...
    'NumPhaseShifterBits',7,'PropagationSpeed',c);
%% Load Analog Calibration and Perform Digital Calibration
load('16-Mar-2023_15-42_CalibrationData.mat');
% Turn on all the channels and apply the analog calibration phase. Collect
% one set of data from both channels on Pluto.
bf.RxPowerDown(:) = 0;
bf.RxPhase(:) = PhaseCal;
bf.RxGain(:) = calibGainCode;
bf.LatchRxSettings();
rx();
receivedSig_HW = rx();
receivedFFT = fft(receivedSig_HW);
PlutoGainDifference = mag2db(max(abs(receivedFFT(:,1)))./ ...
    max(abs(receivedFFT(:,2))));
rx.GainChannel1 = rx.GainChannel1 + round(PlutoGainDifference);

% Find the digital calibration phase
phase = deg2rad(-180 : 180);
st_vec = [ones(size(phase)); exp(1i*phase)];
sig = receivedSig_HW*conj(st_vec);
sigfft = fft(sig);
```

Goal: Establish a Common Design Language and Development Framework

- Create algorithm and device models

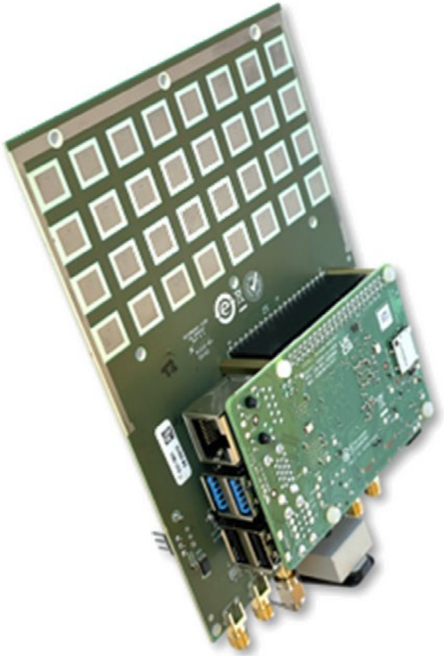
- Simulate

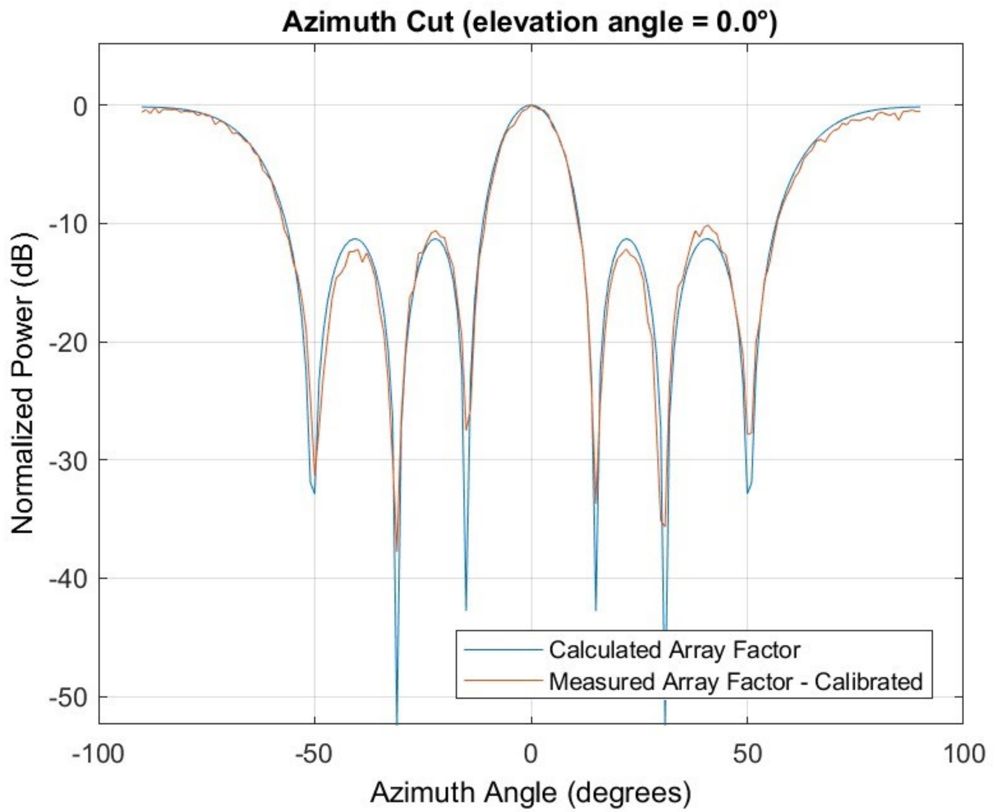




Goal: Establish a Common Design Language and Development Framework

- Create algorithm and device models
- Simulate
- Validate with hardware





Model-Based Design

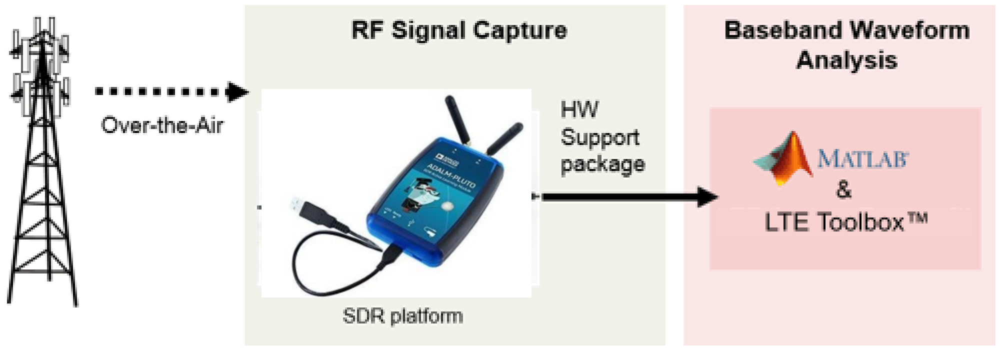
- Models are developed early in the design process to understand what is possible
- Test cases are developed up-front, and the design is verified at every step
- Aka Test Driven Development at the system level

Challenge:

Show how phase array improves performance in a real communications example

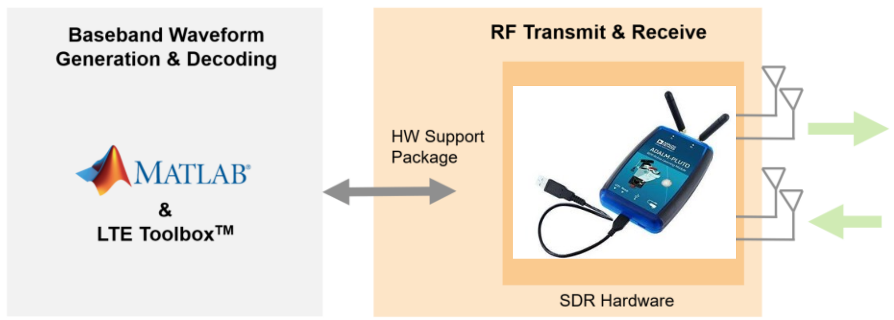
Existing examples

- LTE Receiver Using Software Defined Radio




```
openExample('lte/LTEReceiverUsingSDRExample')
```

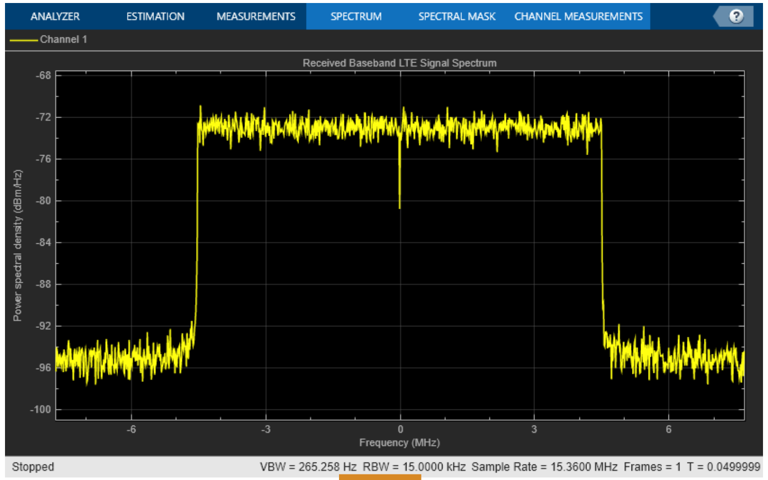
- Image Transmission and Reception Using LTE Waveform and SDR

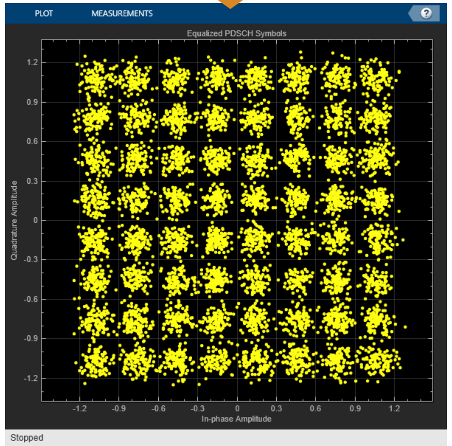



```
openExample('lte/SDRImageTransmissionReceptionUsingLTEWaveformExample')
```

Image Transmission and Reception Using LTE Waveform and SDR

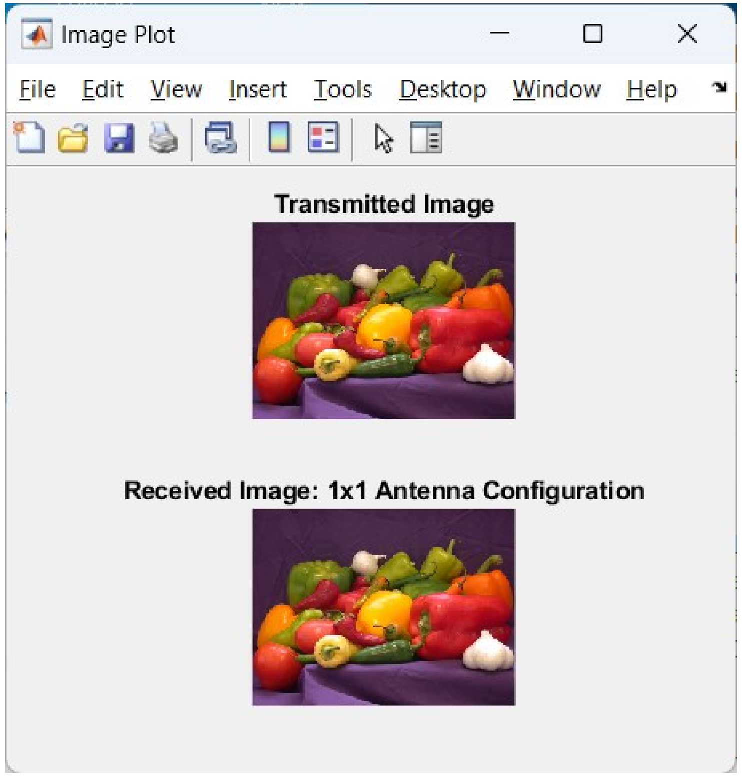
LTE encode



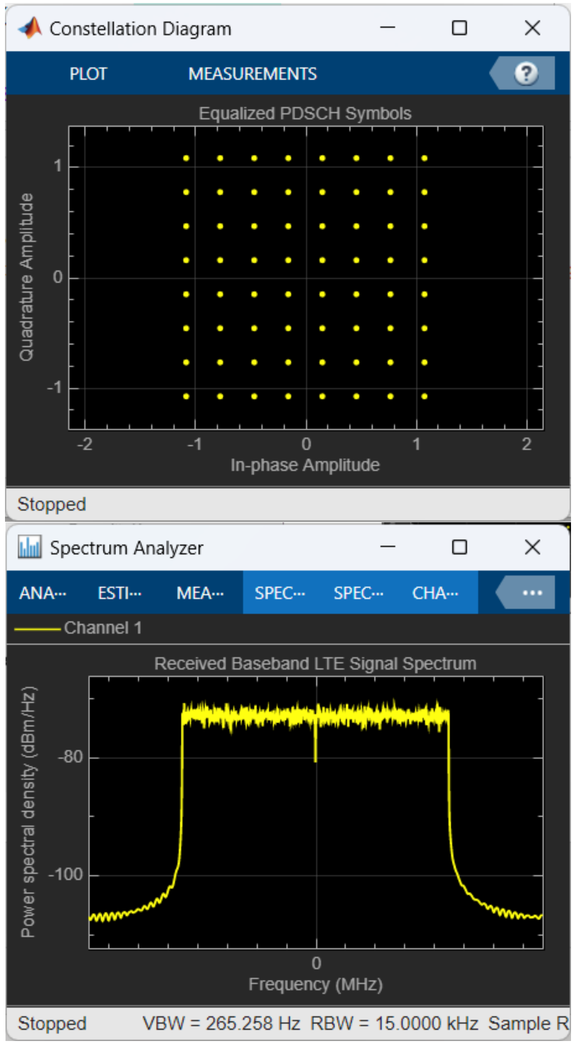


LTE decode

Example progress

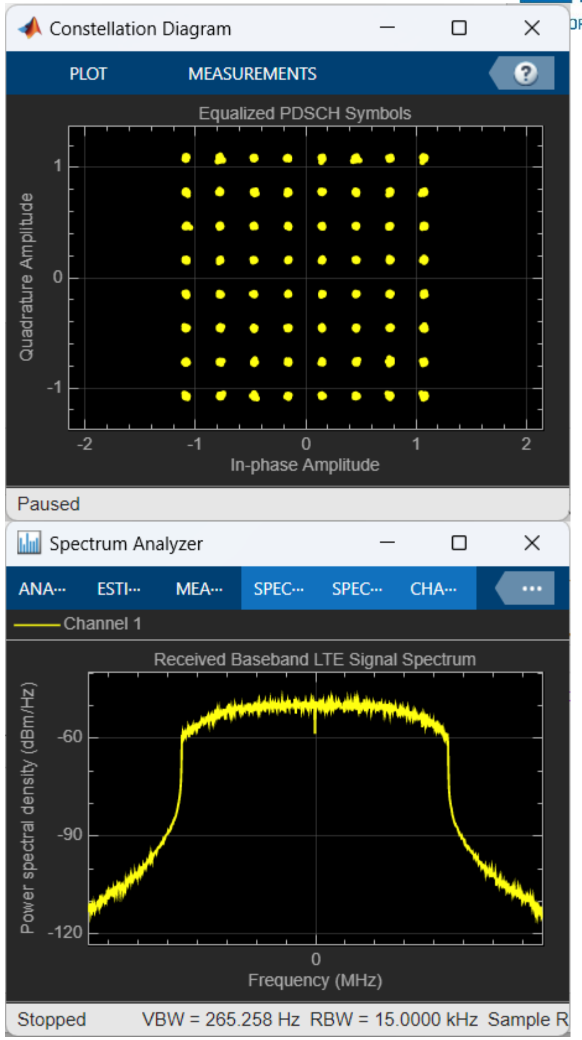


Simulation, no impairments



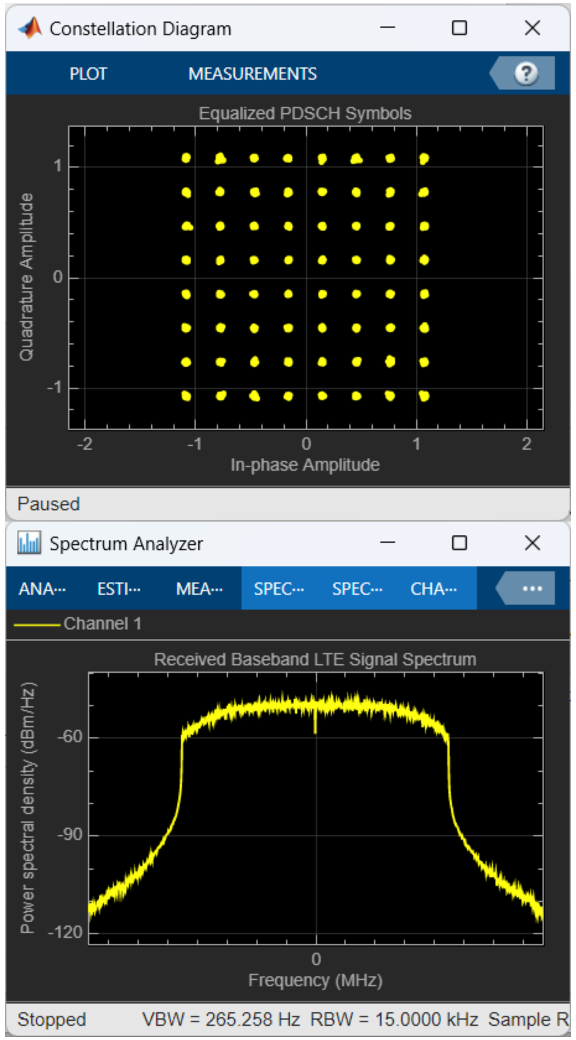
EVM peak = 0.000%
EVM RMS = 0.000%
Bit Error Rate (BER) = 0.00000.
Number of bit errors = 0.
Number of transmitted bits = 1179648.

Pluto SDR, Wire



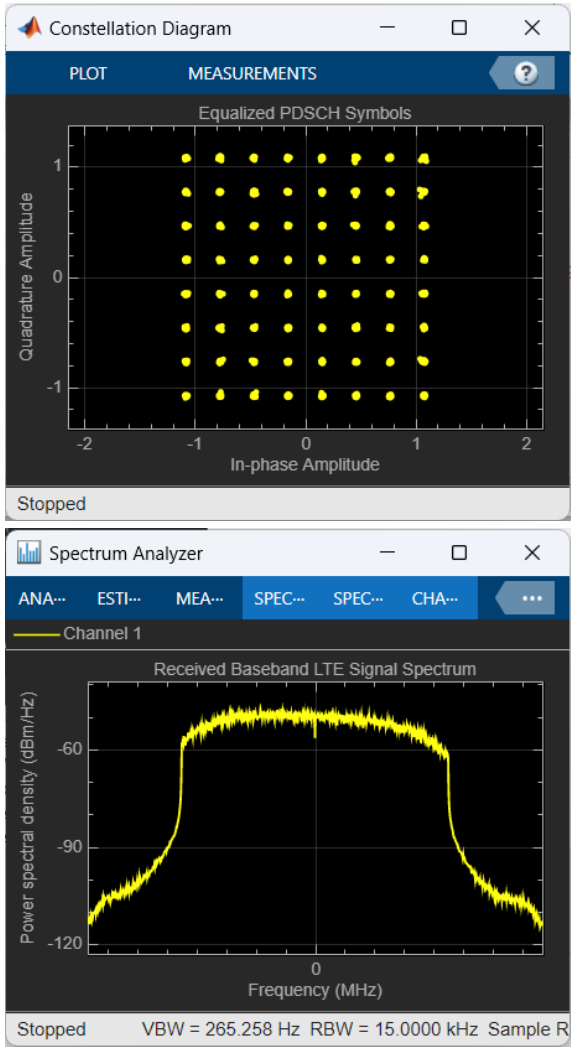
EVM peak = 6.920%
EVM RMS = 0.831%
Bit Error Rate (BER) = 0.00000.
Number of bit errors = 0.
Number of transmitted bits = 1179648.

Pluto SDR, Wire



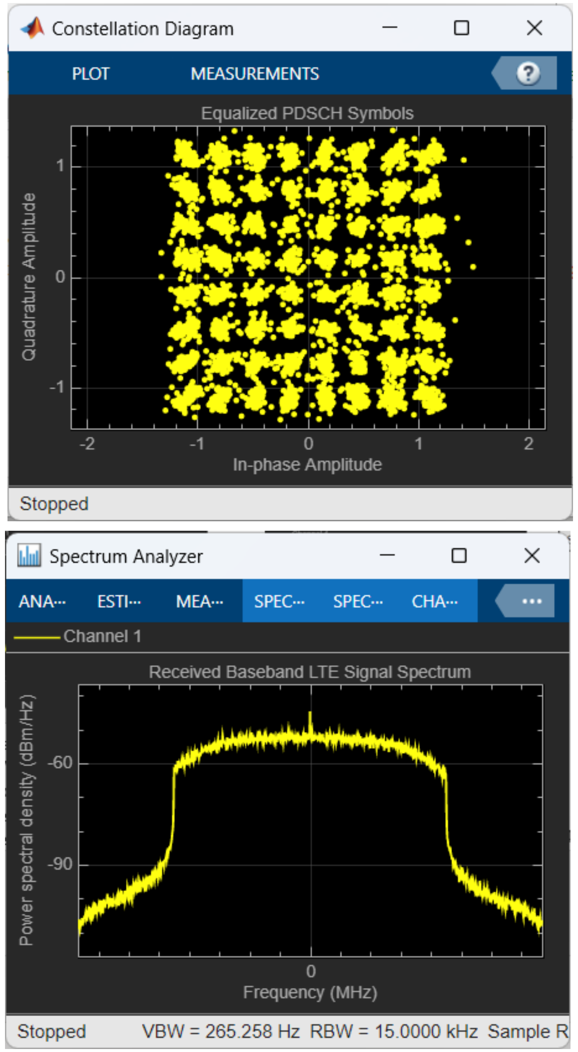
EVM peak = 6.920%
EVM RMS = 0.831%
Bit Error Rate (BER) = 0.00000.
Number of bit errors = 0.
Number of transmitted bits = 1179648.

Pluto SDR, Antenna (2cm)



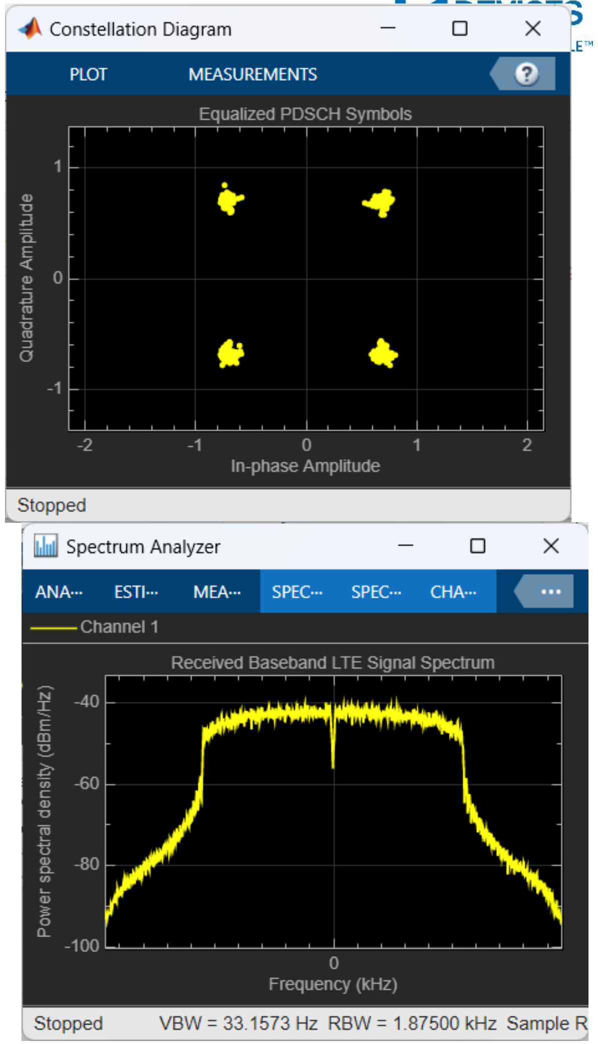
EVM peak = 6.725%
EVM RMS = 0.785%
Bit Error Rate (BER) = 0.00000.
Number of bit errors = 0.
Number of transmitted bits = 1179648.

Pluto SDR, Antenna (1m)



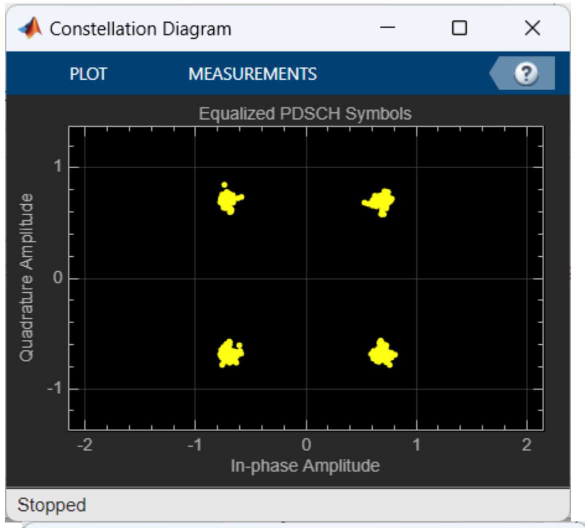
EVM peak = 316.112%
EVM RMS = 25.803%
Bit Error Rate (BER) = 0.00136.
Number of bit errors = 1605.
Number of transmitted bits = 1179648.

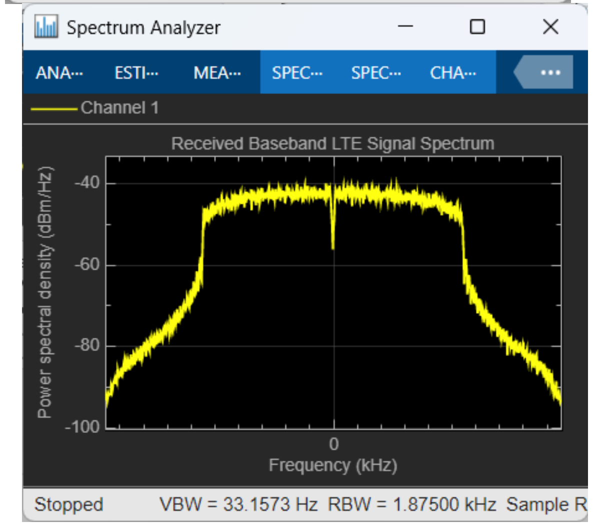
Pluto SDR, Antenna (1m)



EVM peak = 87.036%
EVM RMS = 5.304%
Bit Error Rate (BER) = 0.00000.
Number of bit errors = 0.
Number of transmitted bits = 1179648.

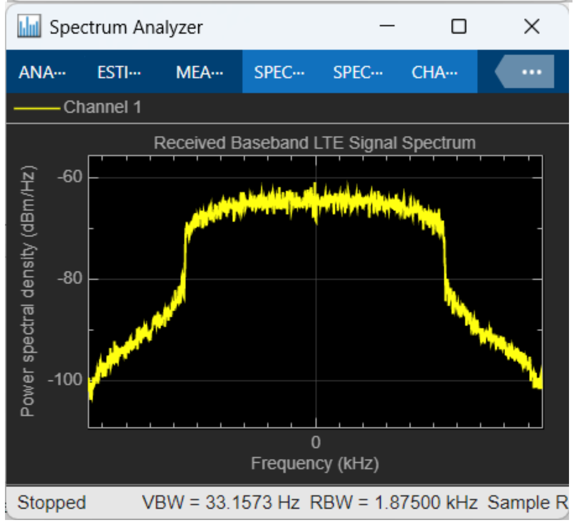
Pluto SDR, Antenna (1m)





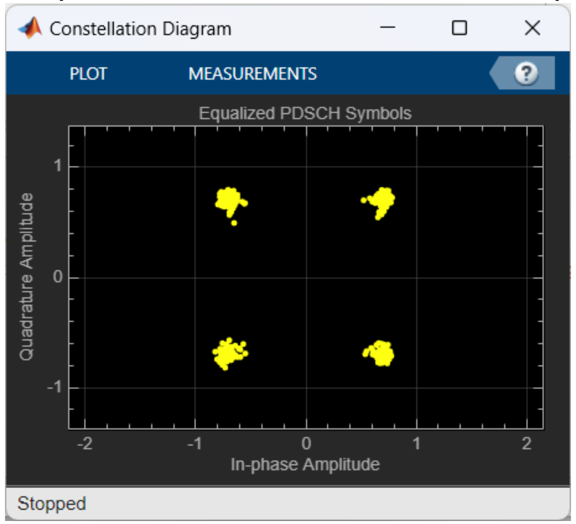
EVM peak = 87.036%
EVM RMS = 5.304%
Bit Error Rate (BER) = 0.00000.
Number of bit errors = 0.
Number of transmitted bits = 1179648.

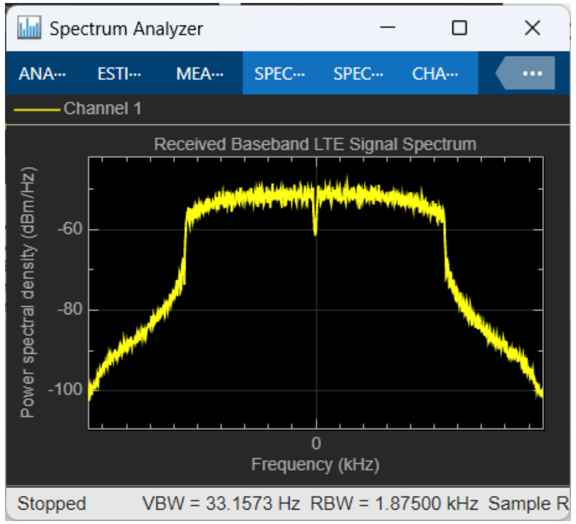
Pluto SDR, Antenna (1m, 6 GHz)



EVM peak = 187.895%
EVM RMS = 20.967%
Bit Error Rate (BER) = 0.00000.
Number of bit errors = 0.
Number of transmitted bits = 1179648.

Pluto SDR, Antenna (1m, 6 GHz)

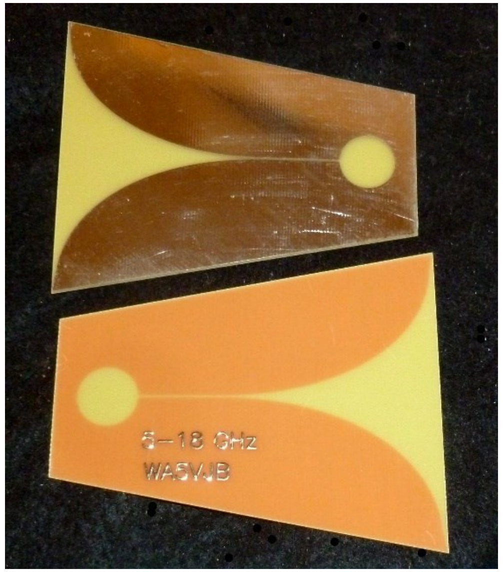




EVM peak = 30.191%
EVM RMS = 5.374%
Bit Error Rate (BER) = 0.00000.
Number of bit errors = 0.
Number of transmitted bits = 1179648..

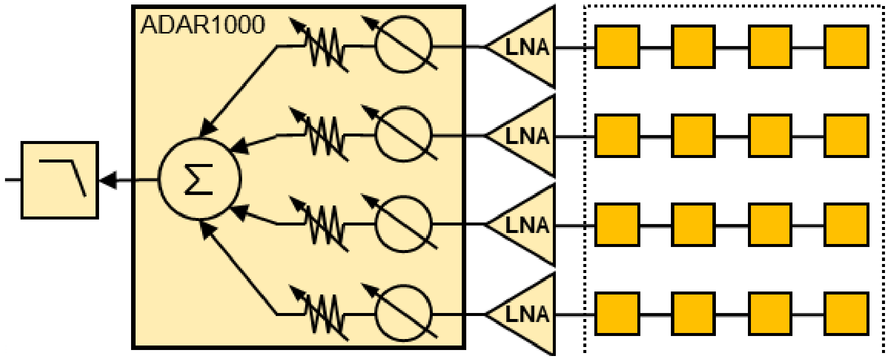
MathWorks
ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

Kent Electronics



https://www.wa5vjb.com/products5.html

Phaser Rx Path



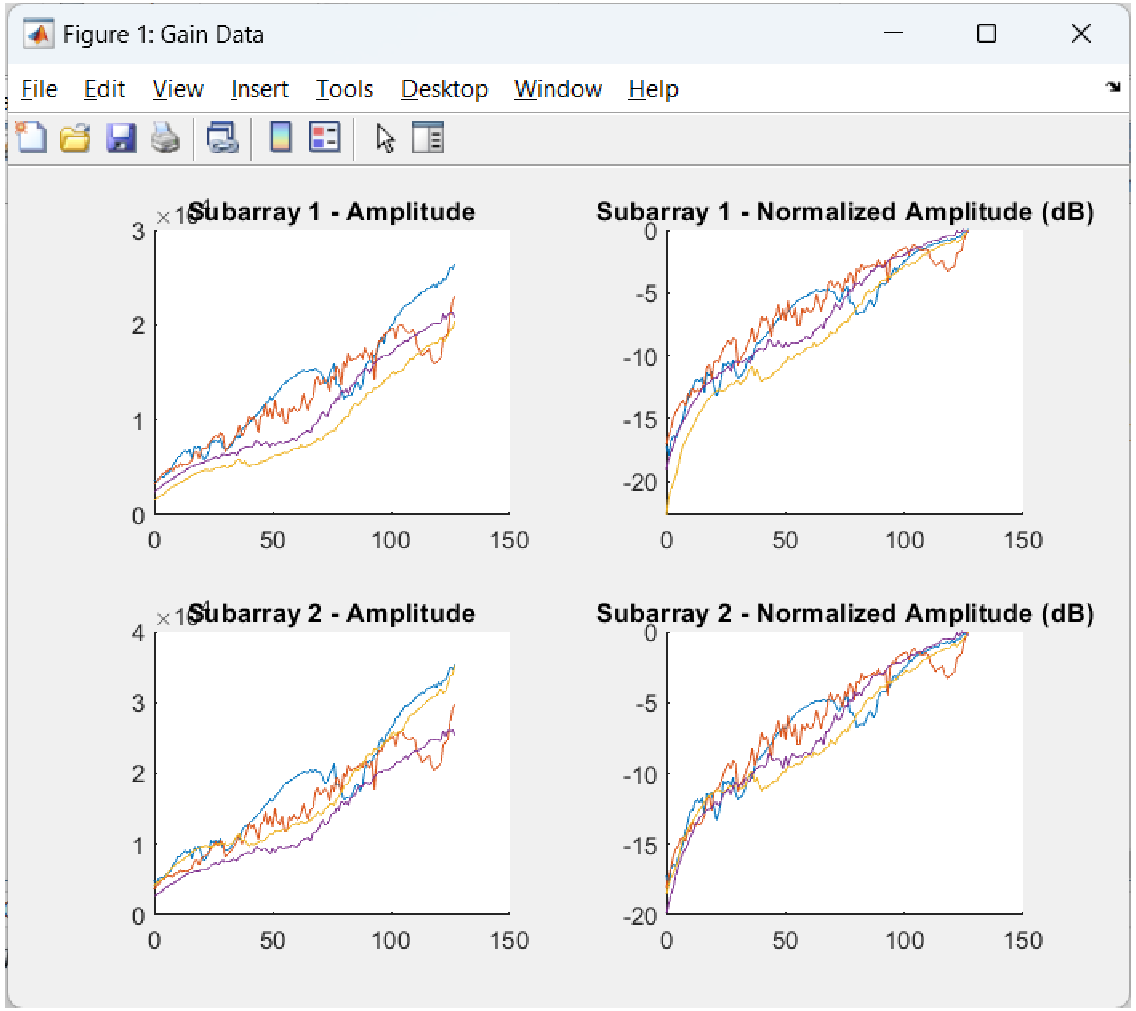
- No spec for channel to channel match of gain or phase in:
  - PCB antenna pattern
  - ADL8107 LNA

GAIN	21.5	24	dB

  - ADAR1000

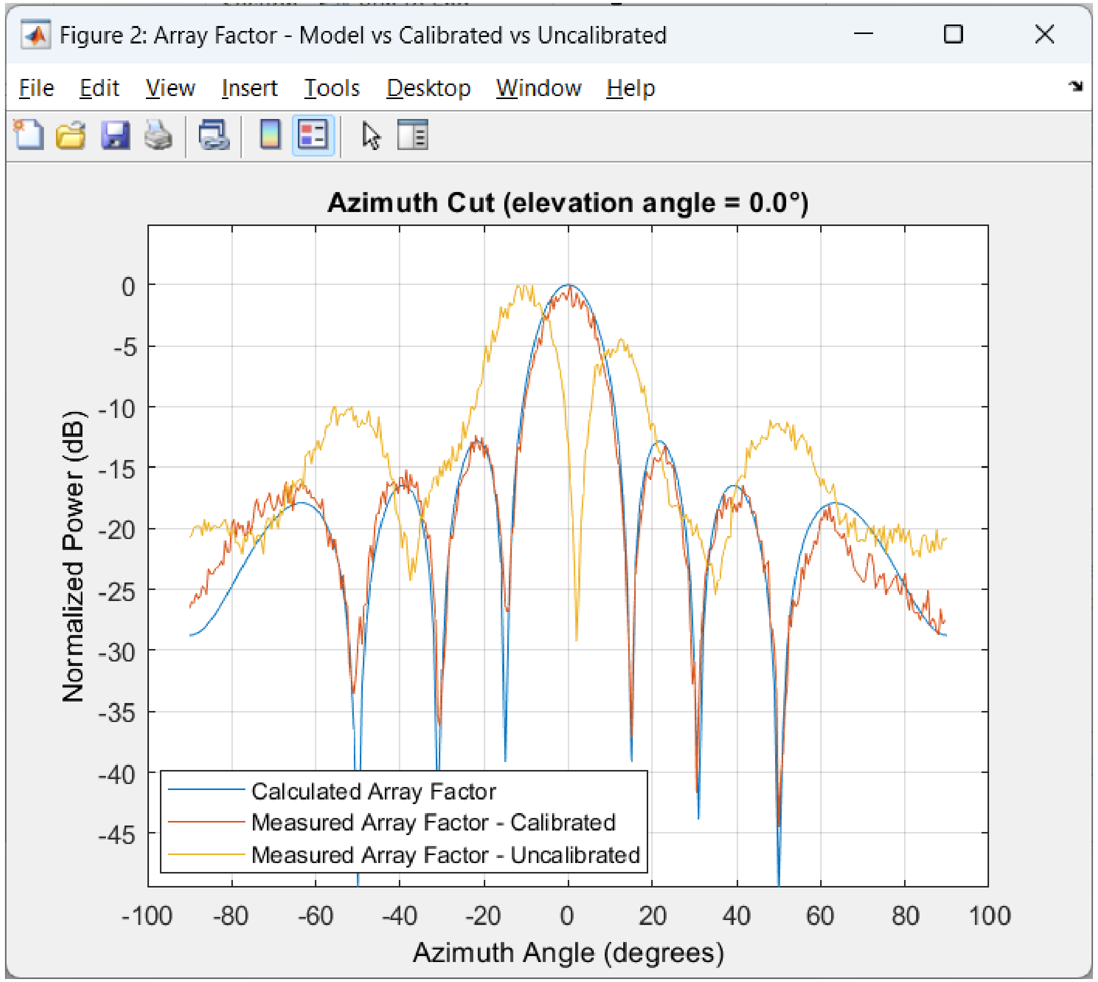
Maximum Single Channel Gain[3] 9.5 GHz	Nominal bias setting	10	dB

- Everything needs calibration at system level



Calibration

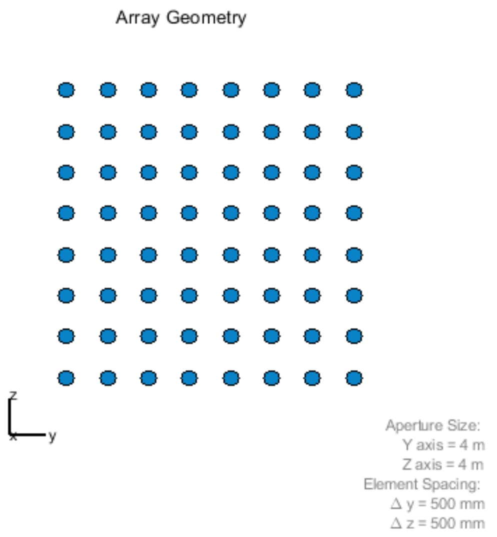
- Measure in free space
  - Be aware of reflections
- Compare to theory, see if things correlate

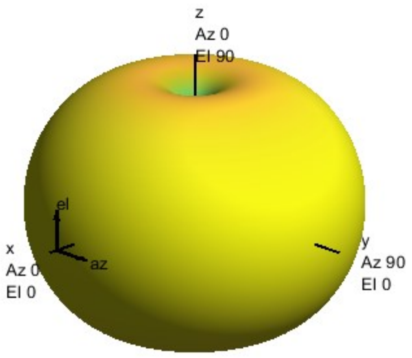


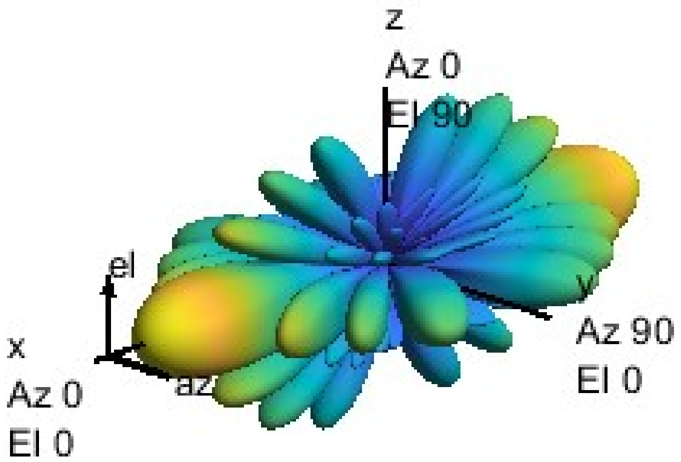
Antenna Array Design and Analysis

- Derive the array pattern from array factor and element pattern

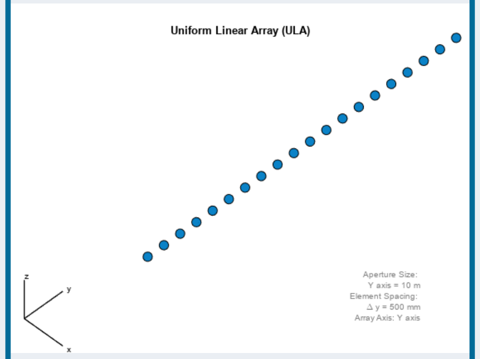
Array Factor × Element Pattern = Array Pattern



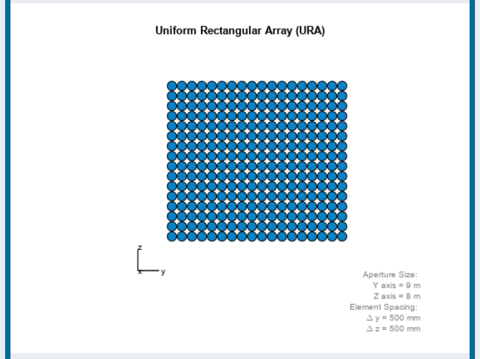




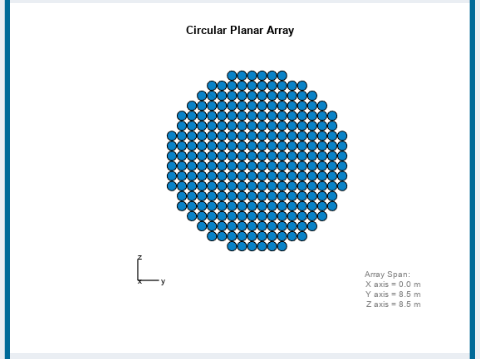
Model arrays with predefined or custom geometries



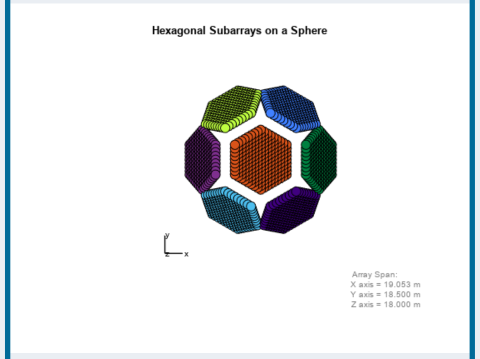
Uniform Linear Array



Uniform Rectangular Array



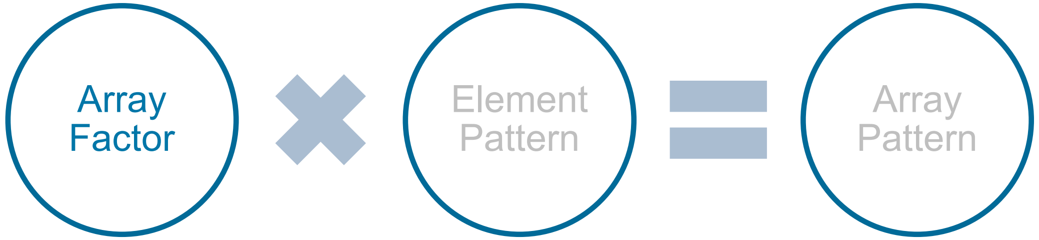
Circular Planar Array

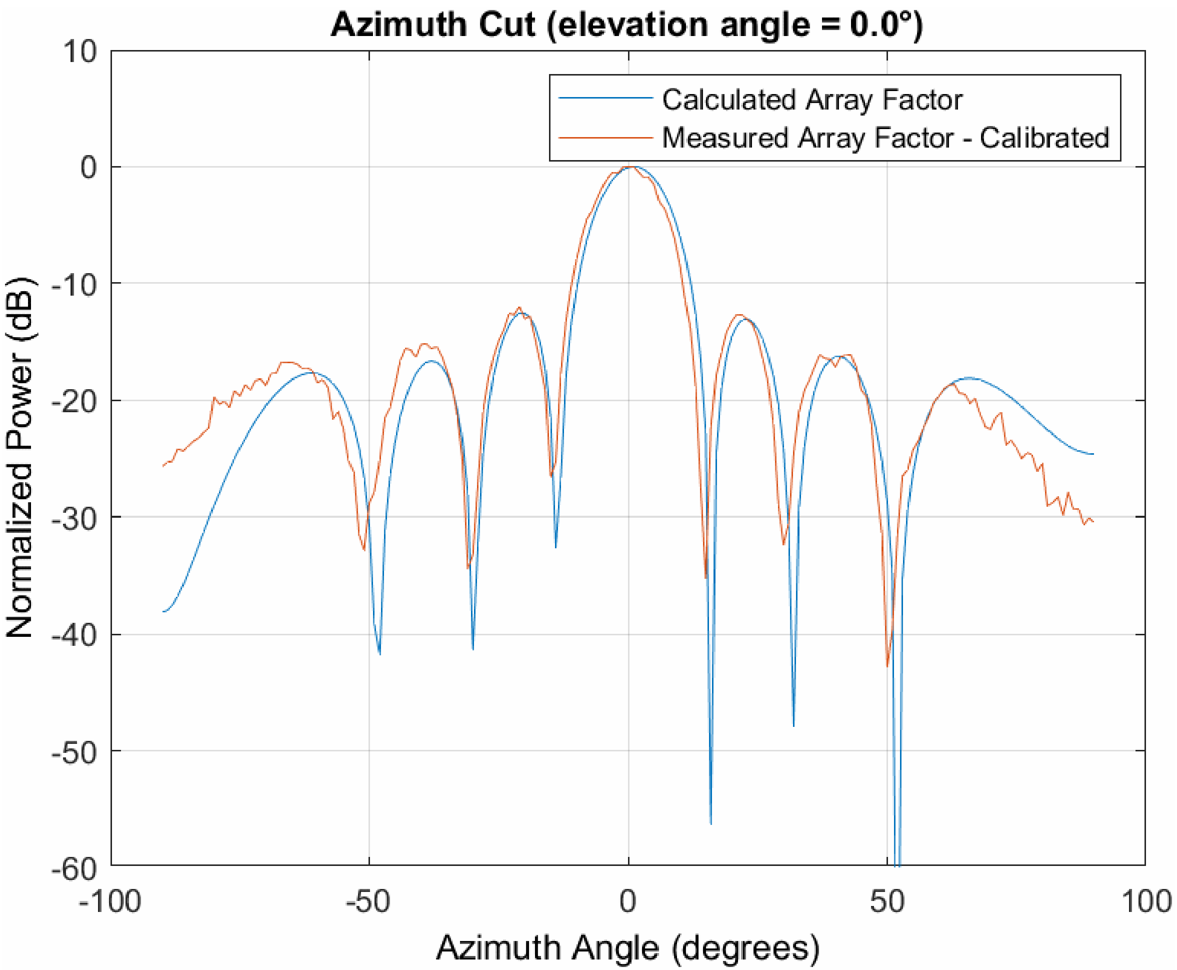


Custom Array Geometry

Array Factor ✖ Element Pattern = Array Pattern

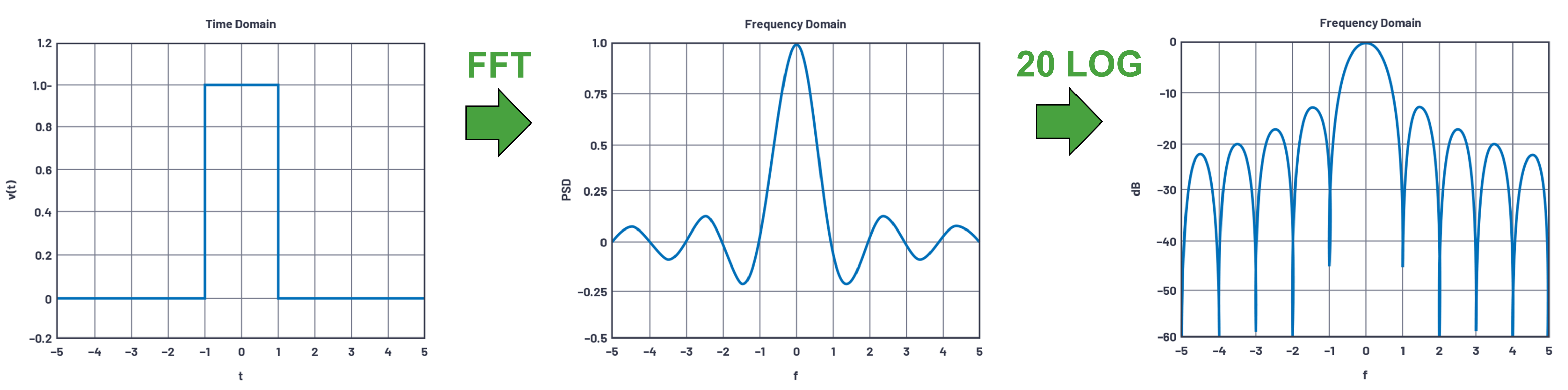
Measure Array Pattern





Sidelobes

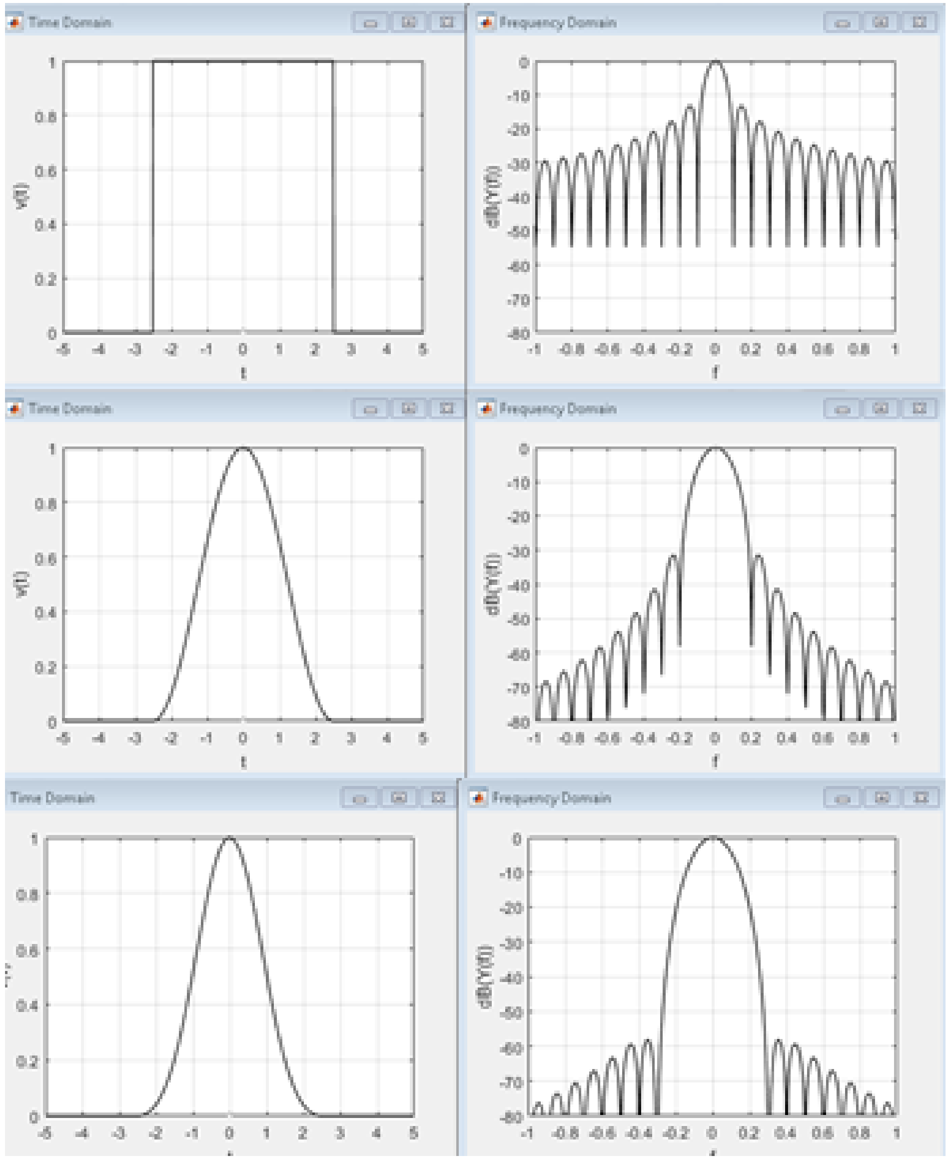
- With all elements at the same gain, we effectively have a box car window.
  - This is analogous to rectangular window FFT



  - Time domain pulse → frequency domain sin(x)/x → first sidelobe -13dBc, etc.
  - As pulse becomes wider…
    - Main lobe narrows
    - Sidelobes move in
    - Sidelobe levels remain unchanged

https://www.analog.com/en/analog-dialogue/articles/phased-array-antenna-patterns-part3.html

Understanding Beam Tapering: Window Functions



Boxcar – 1st sidelobe @ -13dBc
Narrowest main lobe

Hanning – 1st sidelobe < -30dBc
Main lobe broadens

Blackman – Lowest sidelobes
Broadest main lobe

Note: windowing losses not shown in these examples

Sidelobe Control: Beam Tapering

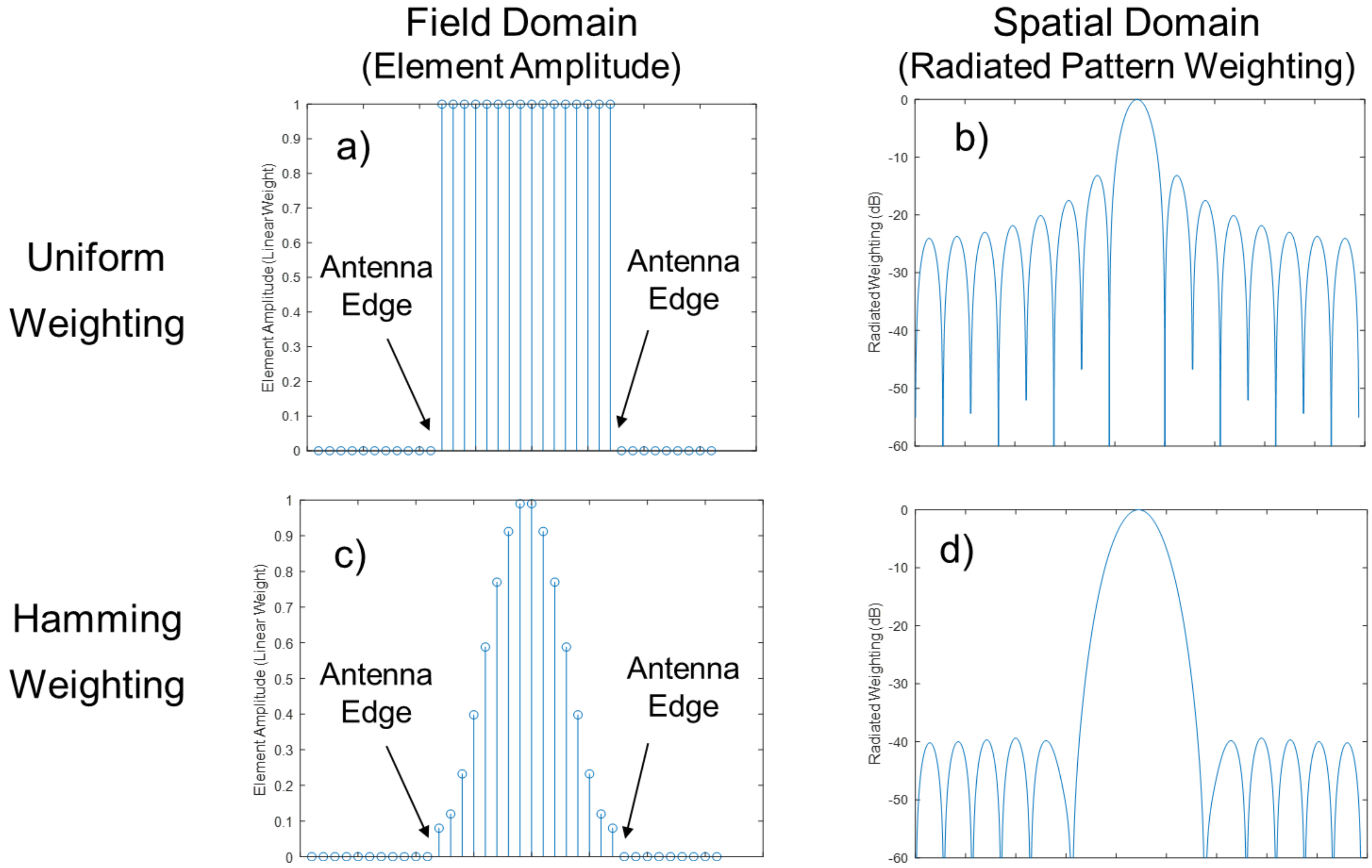
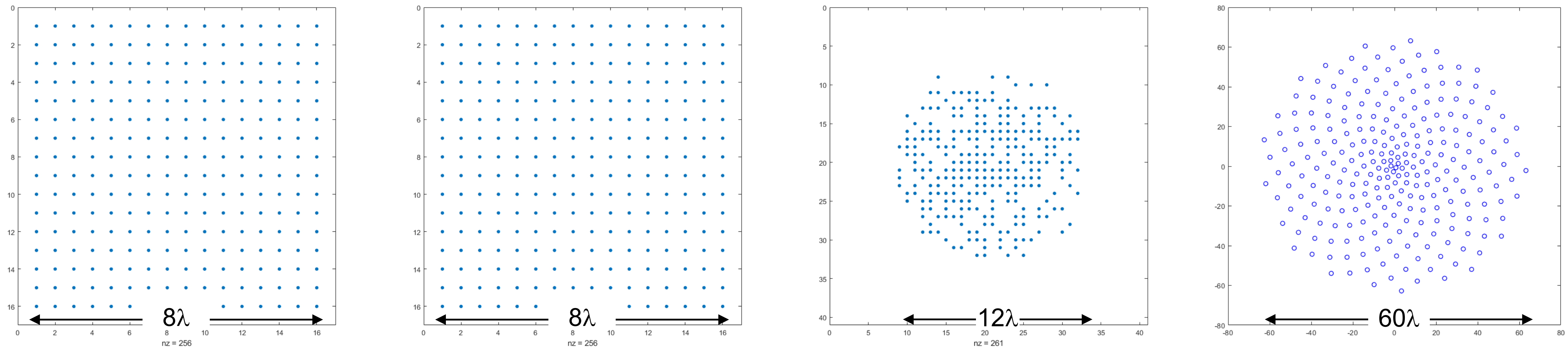
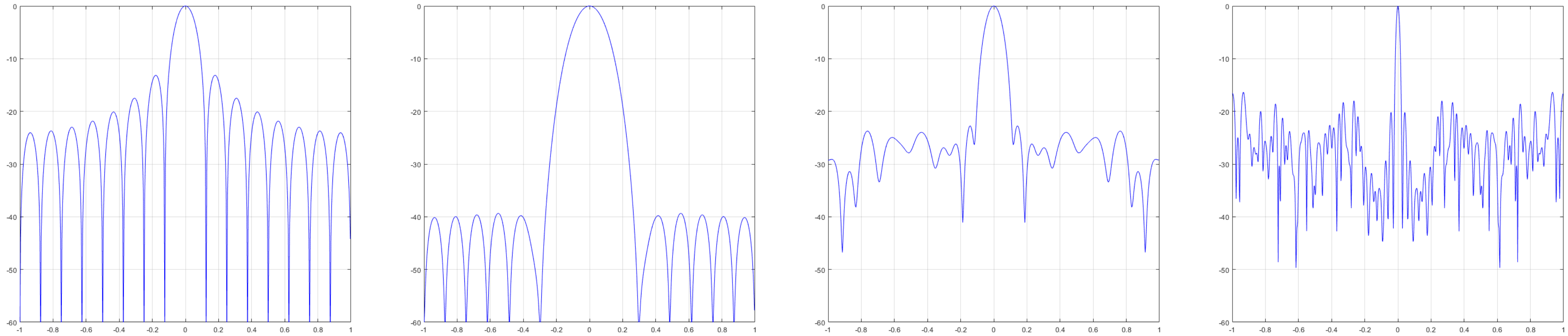


Figure from “Phased Array Antenna Patterns—Part 3: Sidelobes and Tapering”

Sidelobe Control: Comparison of Tapering Methods

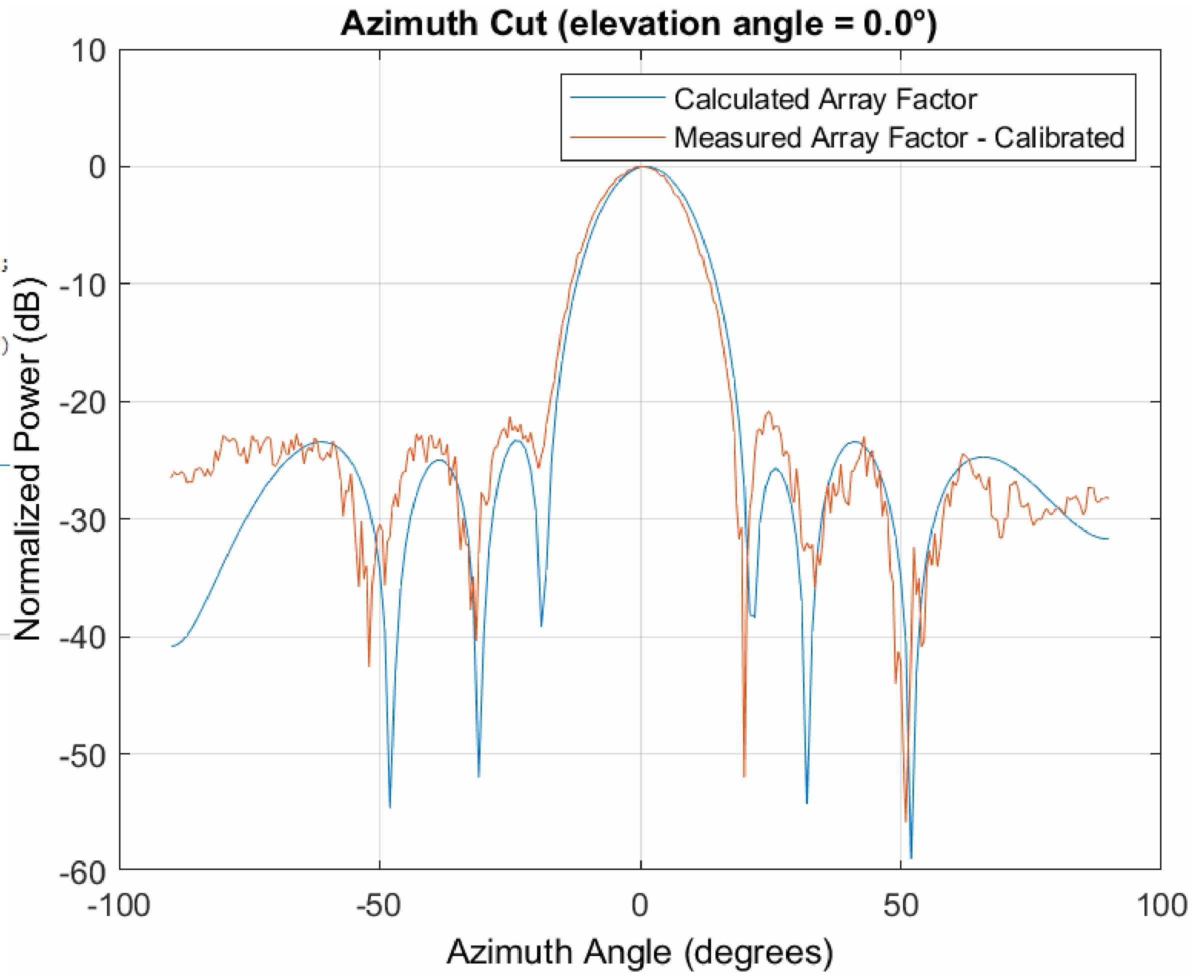


Uniform Illumination | Hamming Amplitude | Hamming Thinned | Spiral



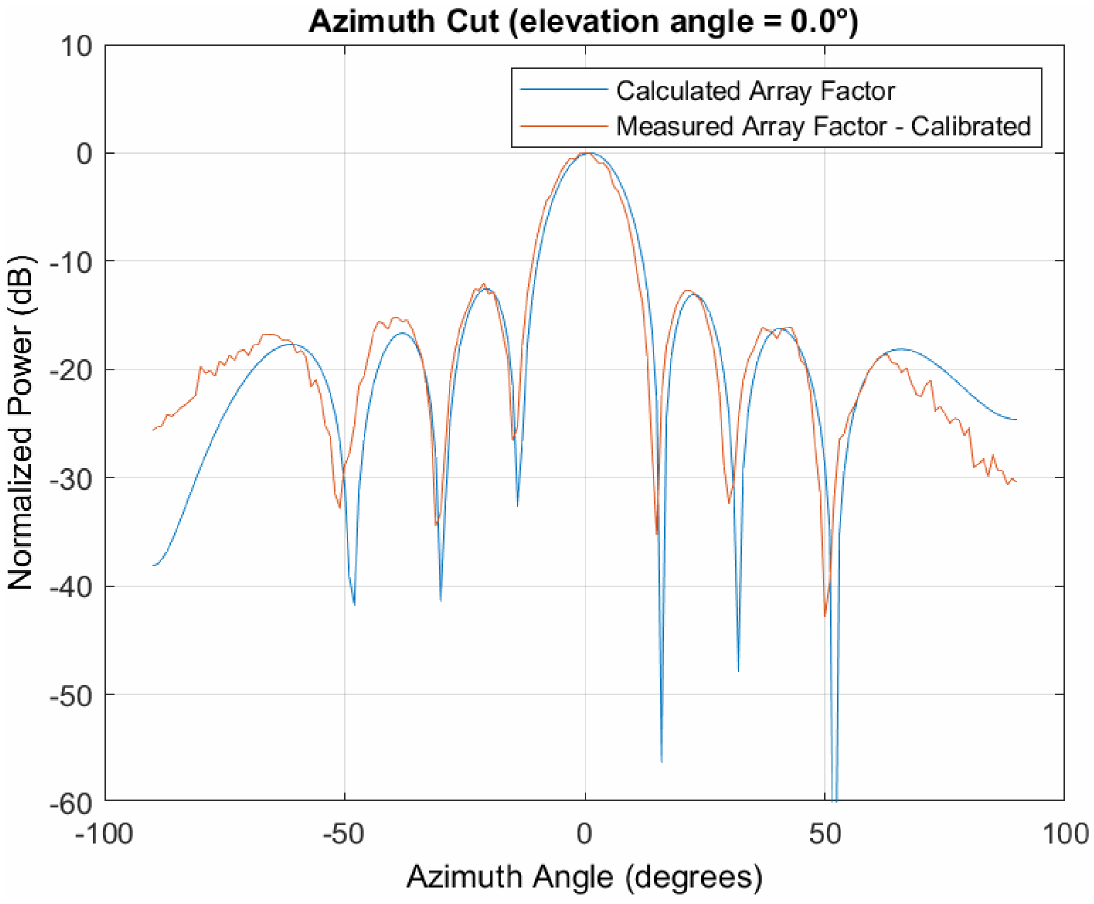
Tapering Example

```
%% Add taper and calibration gain to find new gain control codes
taper_dB = mag2db(taper);

subarray1_TaperGainCal = subarray1_CalibGaindB + taper_dB(1:4);
subarray2_TaperGainCal = subarray2_CalibGaindB + taper_dB(5:8);
subarray1_TaperGainCal = subarray1_TaperGainCal - max(subarray1_TaperGainCal);
subarray2_TaperGainCal = subarray2_TaperGainCal - max(subarray2_TaperGainCal);
load('16-Mar-2023_15-31_GainProfile.mat');

calibGainCode = zeros(1,8);
for nch = 1 : 4

    xp = subarray1_TaperGainCal(nch);
    calibGainCode(nch) = round(interp1(subArray1_NormalizedGainProfile(:,nch),gaincode,xp));

    xp = subarray2_TaperGainCal(nch);
    calibGainCode(nch+4) = round(interp1(subArray2_NormalizedGainProfile(:,nch),gaincode,xp)

end
calibGainCode(calibGainCode>127) = 127;

%% Collect data
bf.RxGain(:) = calibGainCode;
bf.RxAttn(:) = 0;
bf.RxPhase(:) = 0;
bf.RxLNAEnable(:) = true;
bf.RxPowerDown(:) = 0;
```

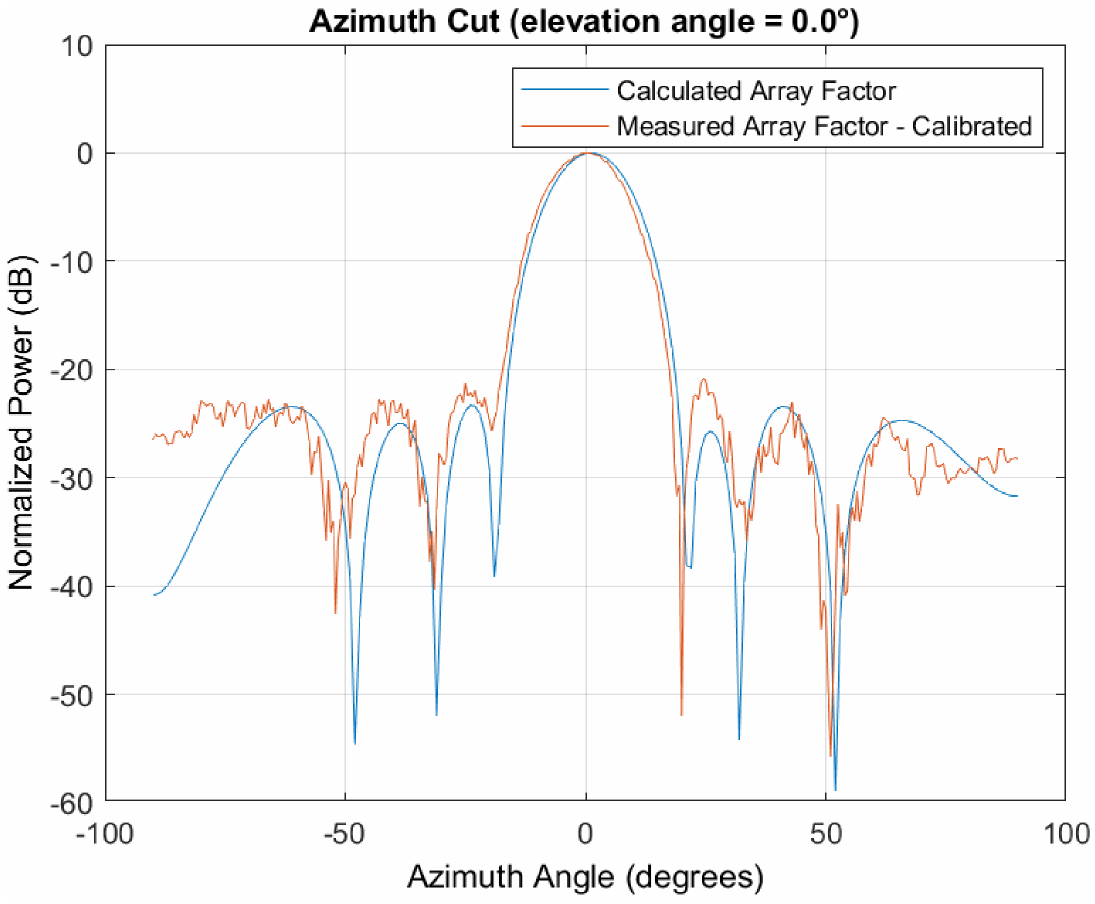


Comparison

Boxcar



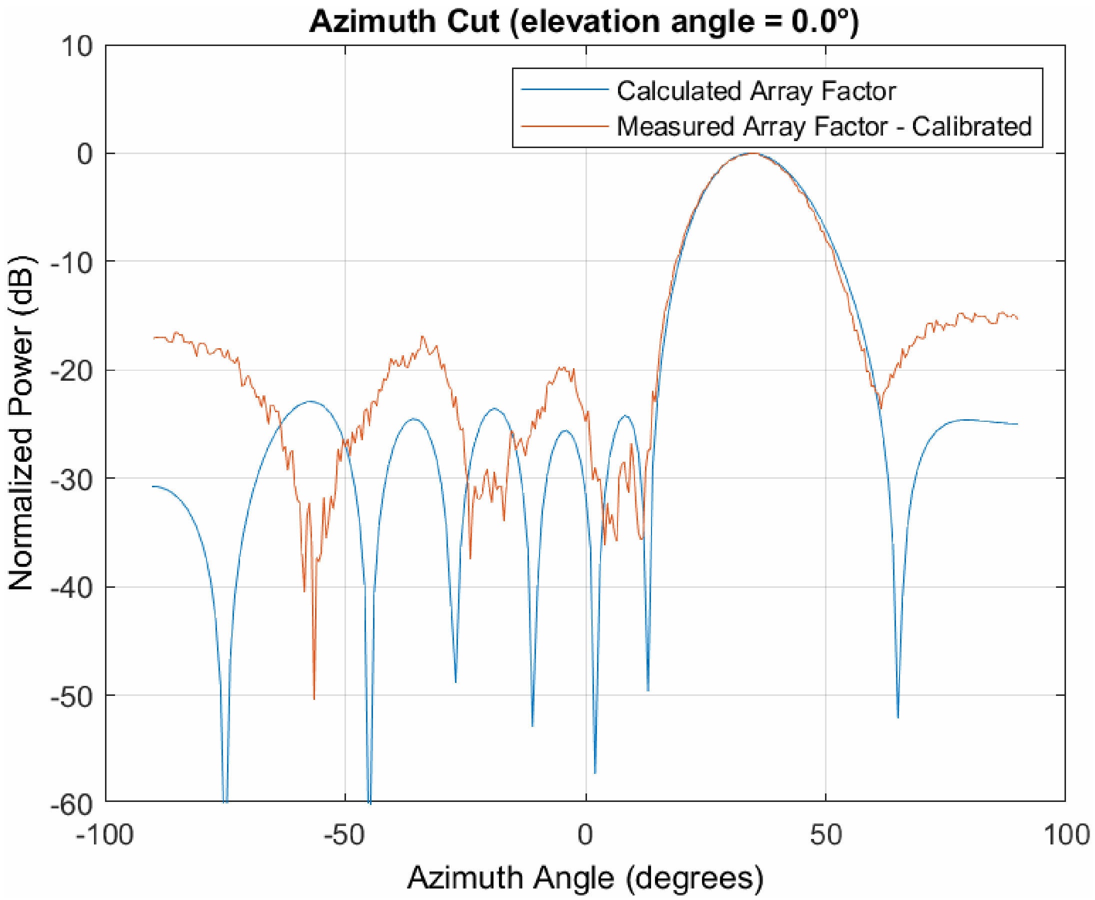
Hamming



Moving the beam?

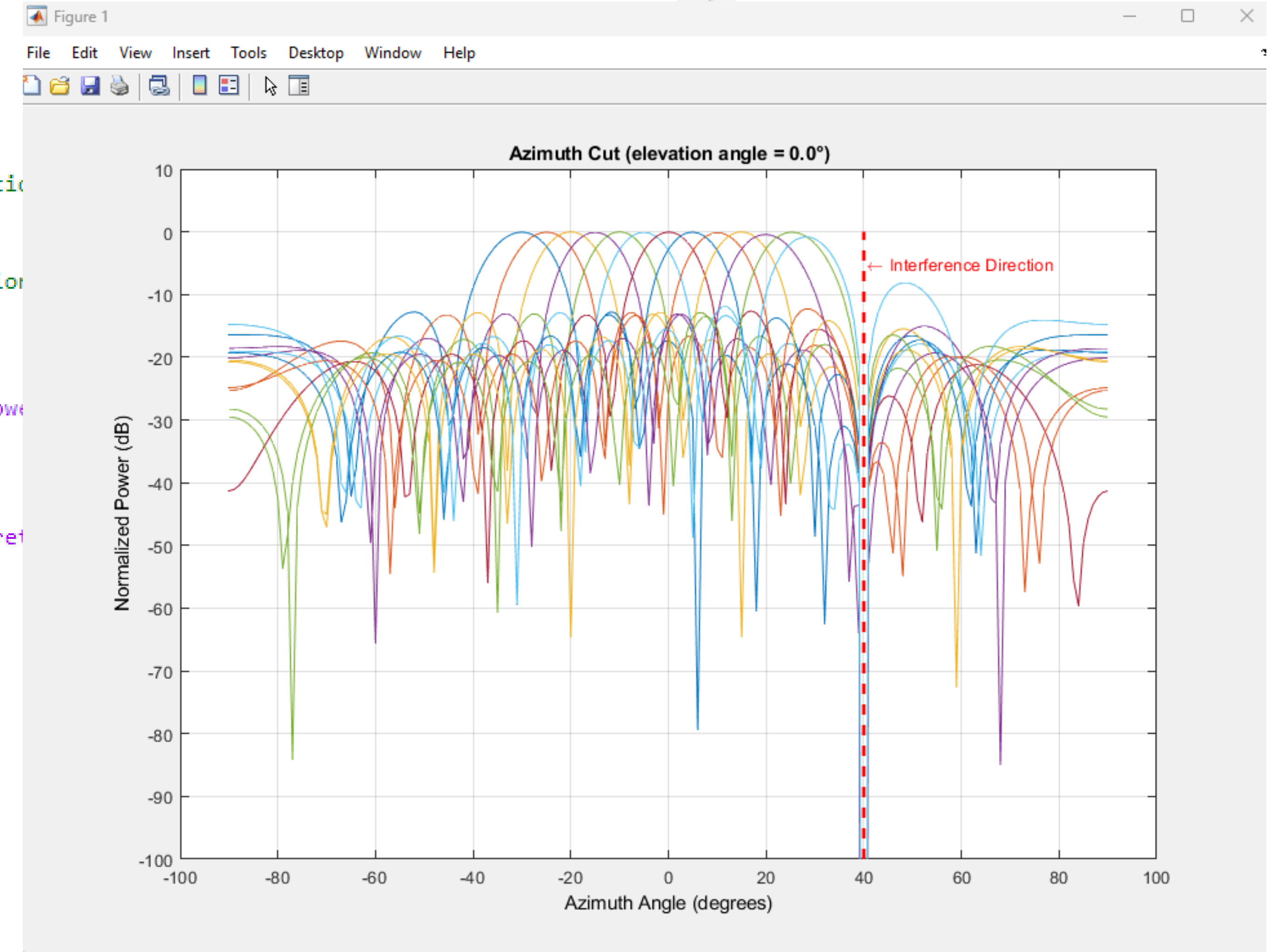
- When the main beam moves, everything moves.
  - All sidelobes
  - All nulls

- Since we have an interferer, how do we move the beam, while still ignoring the interferer
  - Keep a null in a constant place

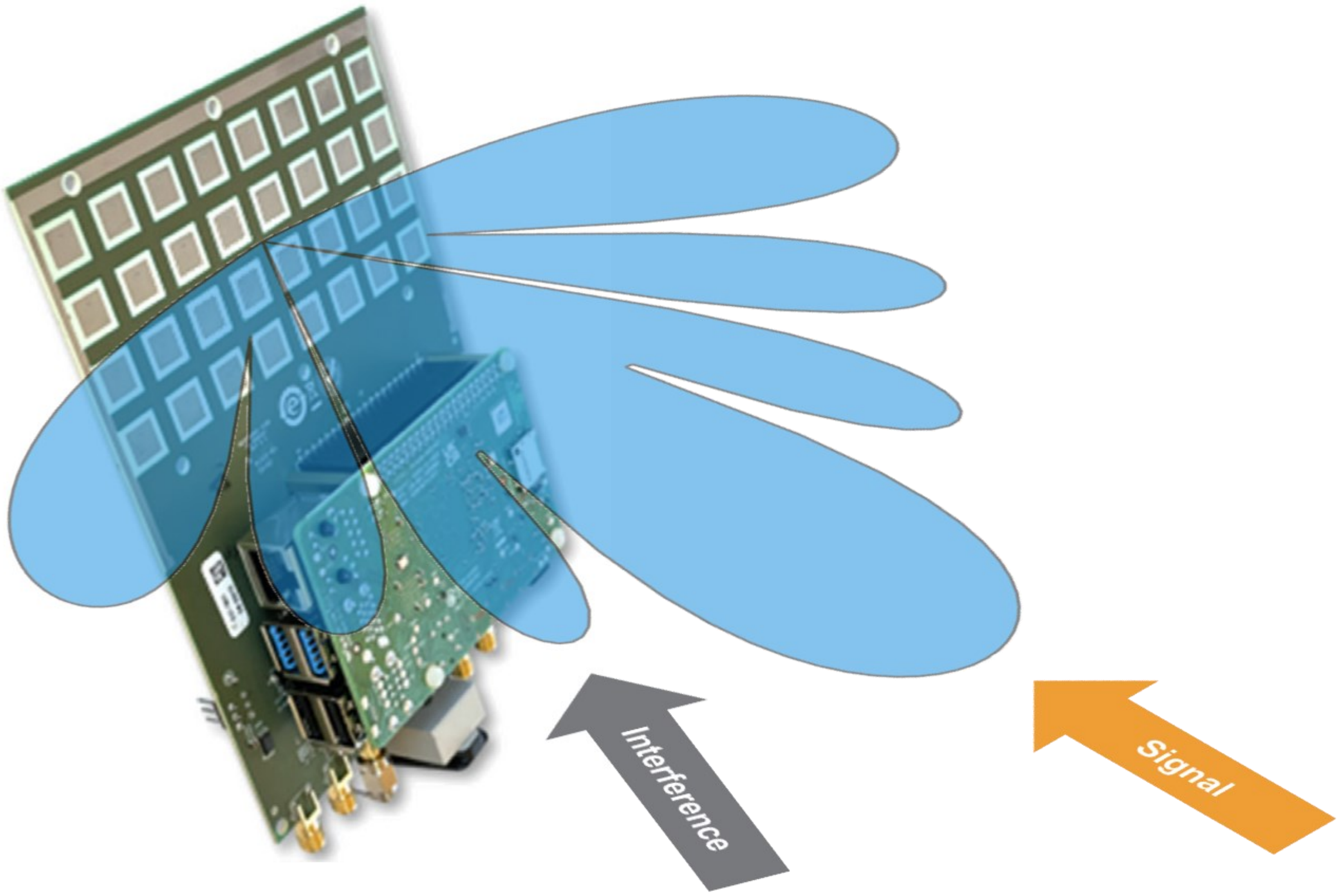


Null Steering

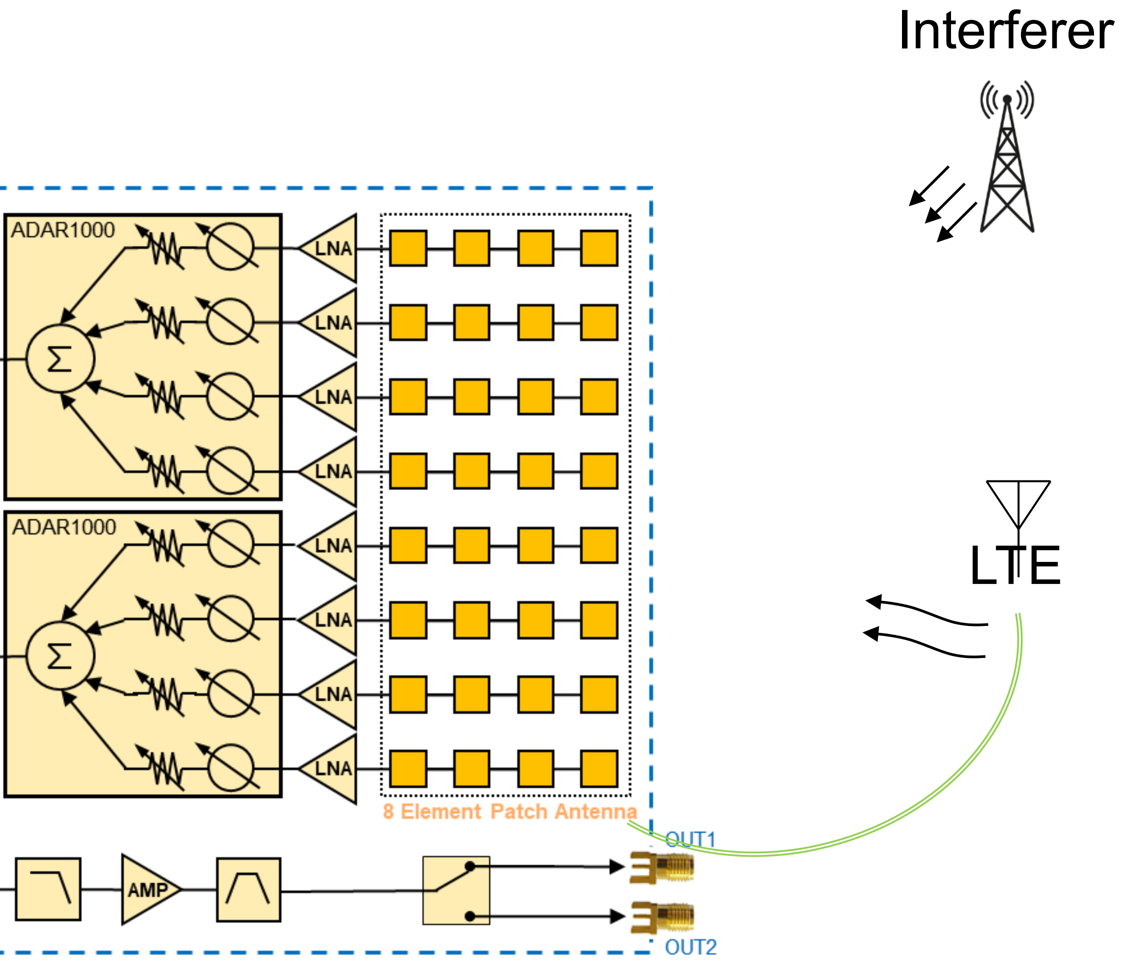
```
% Calculate the steering vector for null directions
wn = steervec(getElementPosition(ula)/lambda,thetaan);

% Calculate the steering vectors for lookout directions
wd = steervec(getElementPosition(ula)/lambda,thetaad);

% Compute the response of desired steering at null directi
rn = wn'*wd/(wn'*wn);

% Sidelobe canceler - remove the response at null directio
w = wd-wn*rn;

% Plot the pattern
pattern(ula,fc,-180:180,0,'PropagationSpeed',c,'Type','powe
    'CoordinateSystem','rectangular','Weights',w);
hold on; legend off;
plot([40 40],[-100 0],'r--','LineWidth',2)
text(40.5,-5,'\leftarrow Interference Direction','Interpret
    'Color','r','FontSize',10)
```

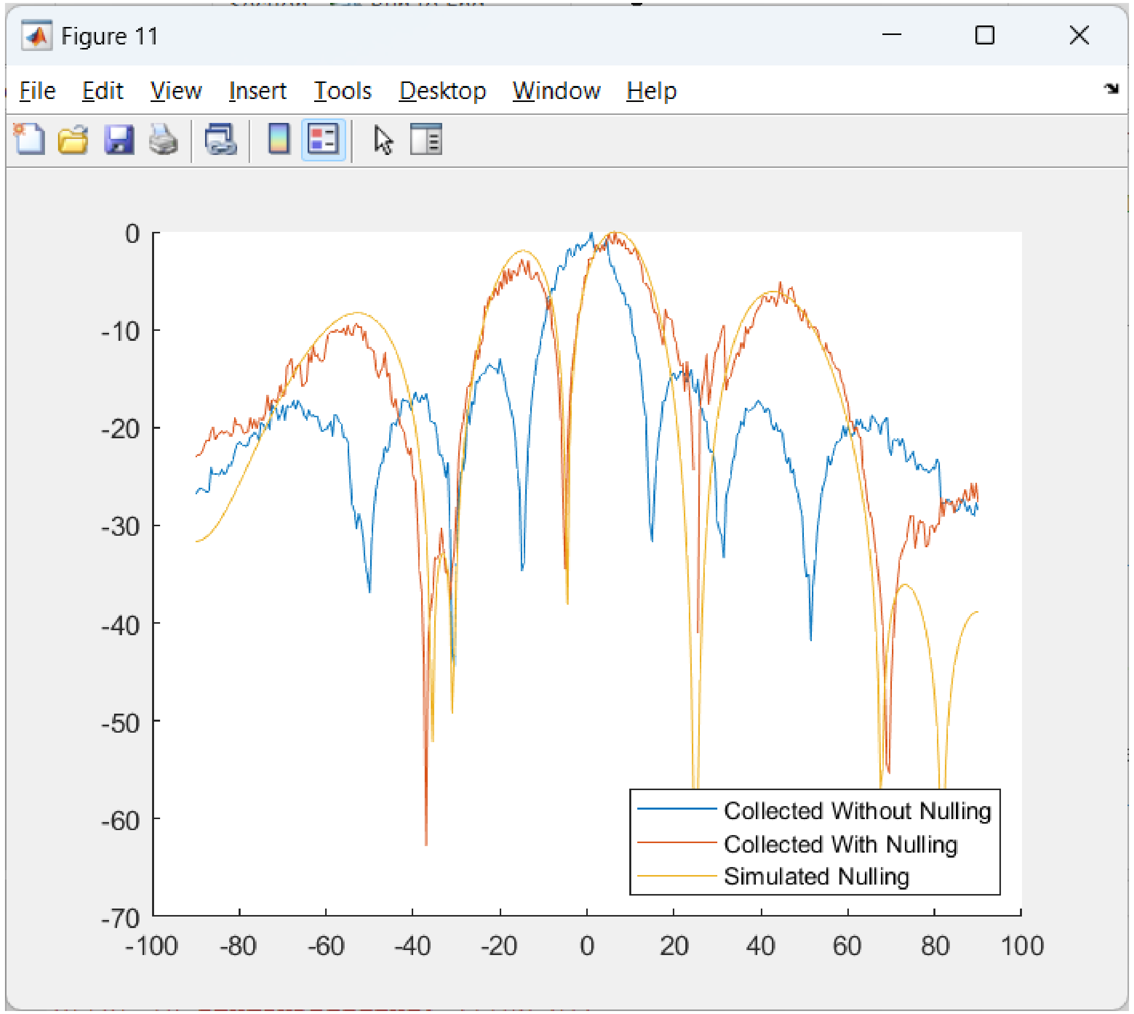


Why do we want to Steer Nulls???



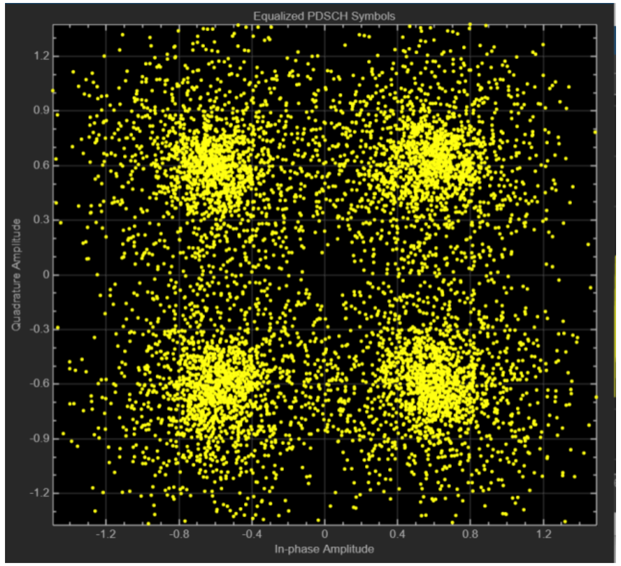
Nulling in action





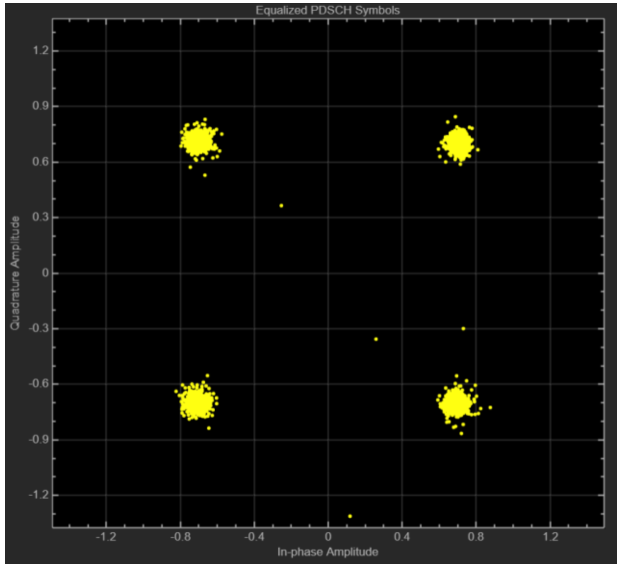
EVM

- No Null Steering



EVM peak = 316.112%
EVM RMS = 40.873%
Bit Error Rate (BER) = 0.00136.
Number of bit errors = 1605.
Number of transmitted bits = 1179648.

- Null Steering On



EVM peak = 30.191%
EVM RMS = 5.187%
Bit Error Rate (BER) = 0.00000.
Number of bit errors = 0.
Number of transmitted bits = 1179648.

Conclusion

- It works
- Simulation matches with real world
- Tested over the air

- Code:
  - https://github.com/mathworks
  - https://wiki.analog.com/resources/eval/user-guides/circuits-from-the-lab/cn0566/matlab
  - https://github.com/analogdevicesinc/RFMicrowaveToolbox

For more information

- Understanding Phased Array Systems and Beamforming
- Brian Douglas
- This video series provides an overview of the concepts related to phased array systems. The series covers the basics of sensor arrays and shows how manipulating the signal to each array element independently can allow for complex beamforming. Throughout the series, see how beamforming is important for many applications, such as multifunction radars and wireless communications.

- [What Are Phased Arrays?]

- [An introduction to Beamforming]

- [Why multichannel beamforming is useful for wireless communication]

- [Why Digital Beamforming Is Useful for Radar]

- [Visualizing Radar Performance with the Ambiguity Function]

https://www.mathworks.com/videos/series/understanding-phased-array-systems-and-beamforming.html

Thanks