Multi-Function Radar Workshop

Part 1: Basic Radar Design

Presenter: Aram Vartanyan, Ph.D. (Application Engineer)

Author: Tony Azar, Ph.D. (Application Engineer)

December 6, 2023

© 2023 The MathWorks, Inc.

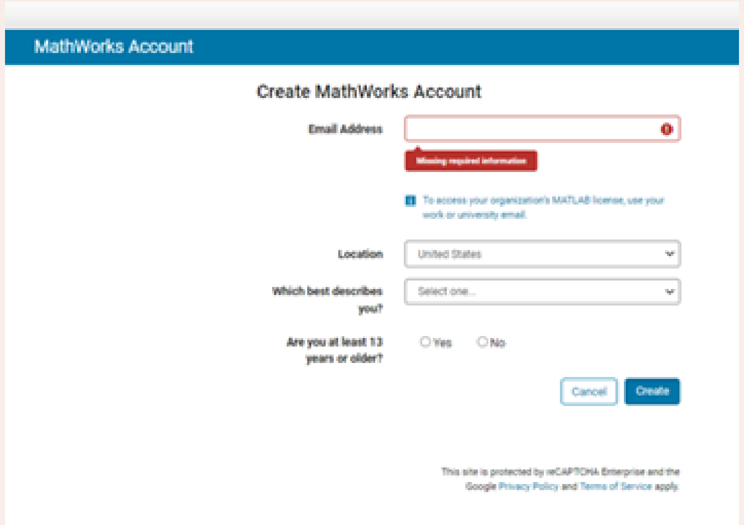
Getting Started

To get started, please follow the step-by-step instructions below to access MATLAB Online

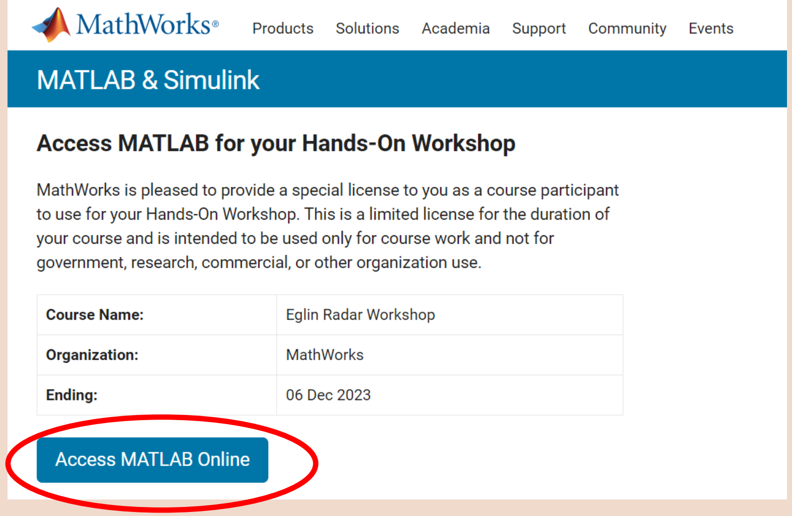
<table>
<tr>
<td>Step 1:
Create a MathWorks account</td>
<td>Go to https://www.mathworks.com

Click on the profile picture located at the top right of the screen.

You will need to provide an email address and some additional information.

If you already have a MathWorks account, you can skip this step.</td>
<td>
</td>
</tr>
<tr>
<td>Step 2:
Access your MATLAB Online Trial</td>
<td>Click on the following link:
https://www.mathworks.com/licensecenter/classroom/4281200/

If prompted, log into your MathWorks account using the same email address that you used to register for the workshop.

Once you're logged in, click on the "Access MATLAB Online" button.</td>
<td>
</td>
</tr>
</table>

Note: Temporary MATLAB Online License is for 30 days!

MathWorks® Products Solutions Academia Support Community Events

MATLAB & Simulink

Access MATLAB for your Hands-On Workshop

MathWorks is pleased to provide a special license to you as a course participant to use for your Hands-On Workshop. This is a limited license for the duration of your course and is intended to be used only for course work and not for government, research, commercial, or other organization use.

Course Name:	Eglin Radar Workshop
Organization:	MathWorks
Ending:	06 Dec 2023

Access MATLAB Online

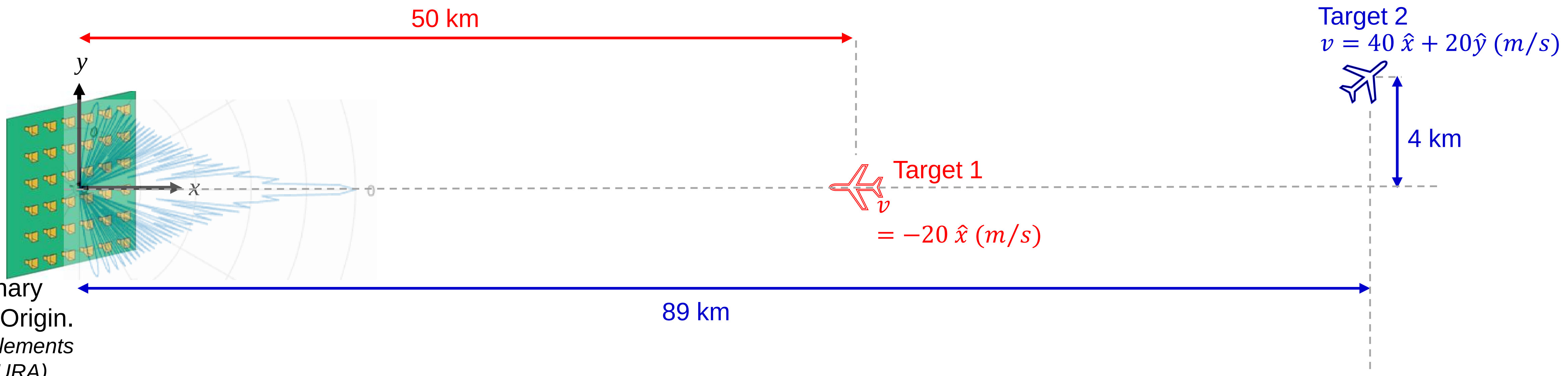
Ending date is displayed incorrectly

You should have access to all the tools for the **next 30 days**

Agenda for Part 1- Basic Radar Design

1) Scenario Overview
2) Block Diagram
3) Waveform Design (Exercise 1 – *Pulse Waveform Analyzer App*)
4) Array Design (Exercise 2 – *Sensor Array Analyzer App*)
5) Radar Designer (Exercise 3 – *Radar Designer App*)
6) Simulation & Results (Exercise 4)

Scenario Overview



Stationary Radar at Origin.
Isotropic Elements (10x10 URA)

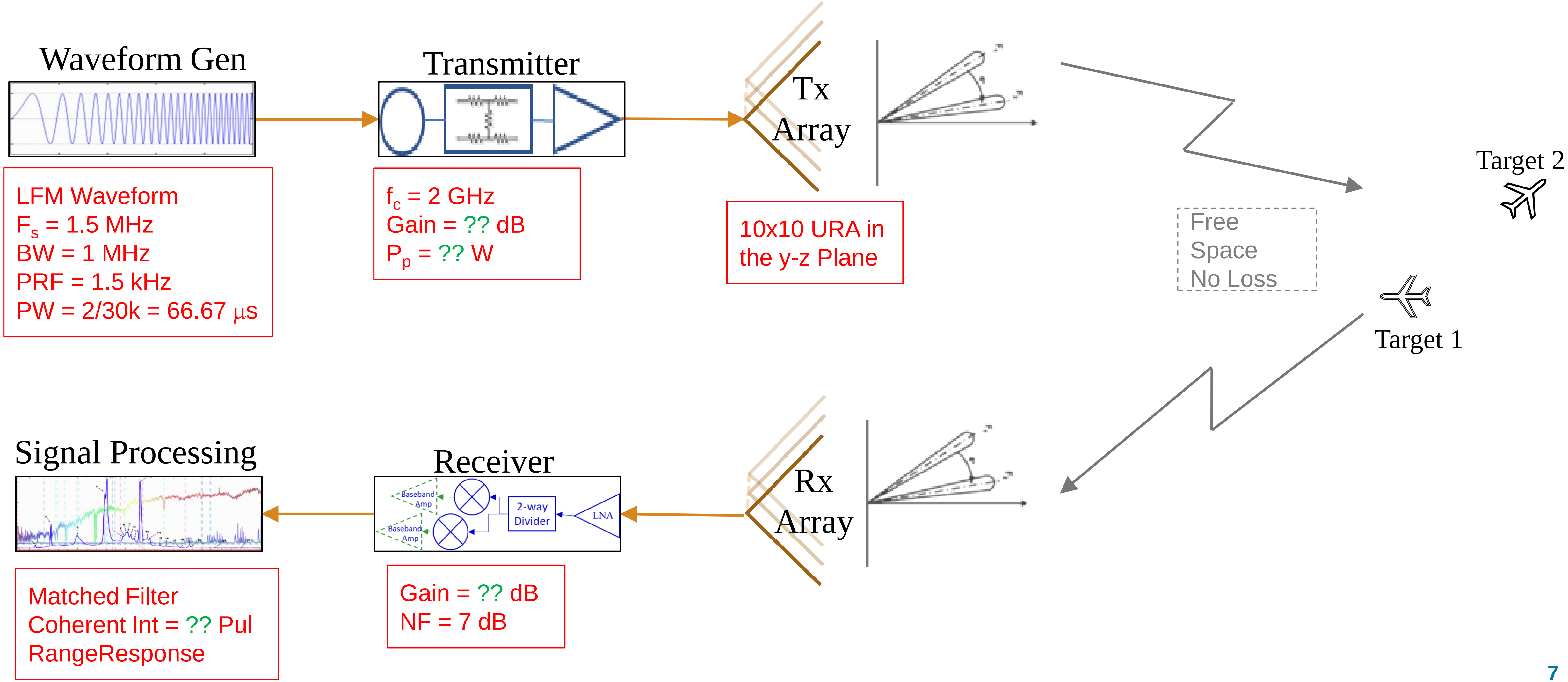
Basic Radar Design

Radar Specs

f_c = 2 GHz
Max Range = 100 km
Ran Res = 150 m

Min RCS = 0.1 m^2
P_d = 0.9 , P_{fa} = 10^{-6}

Block Diagram



Exercise #1 *(Pulse Waveform Analyzer)*

- Use the Pulse Waveform Analyzer to design an LFM waveform
- Export it as a MATLAB Script
- Insert it in the place indicated in Exercise1.m

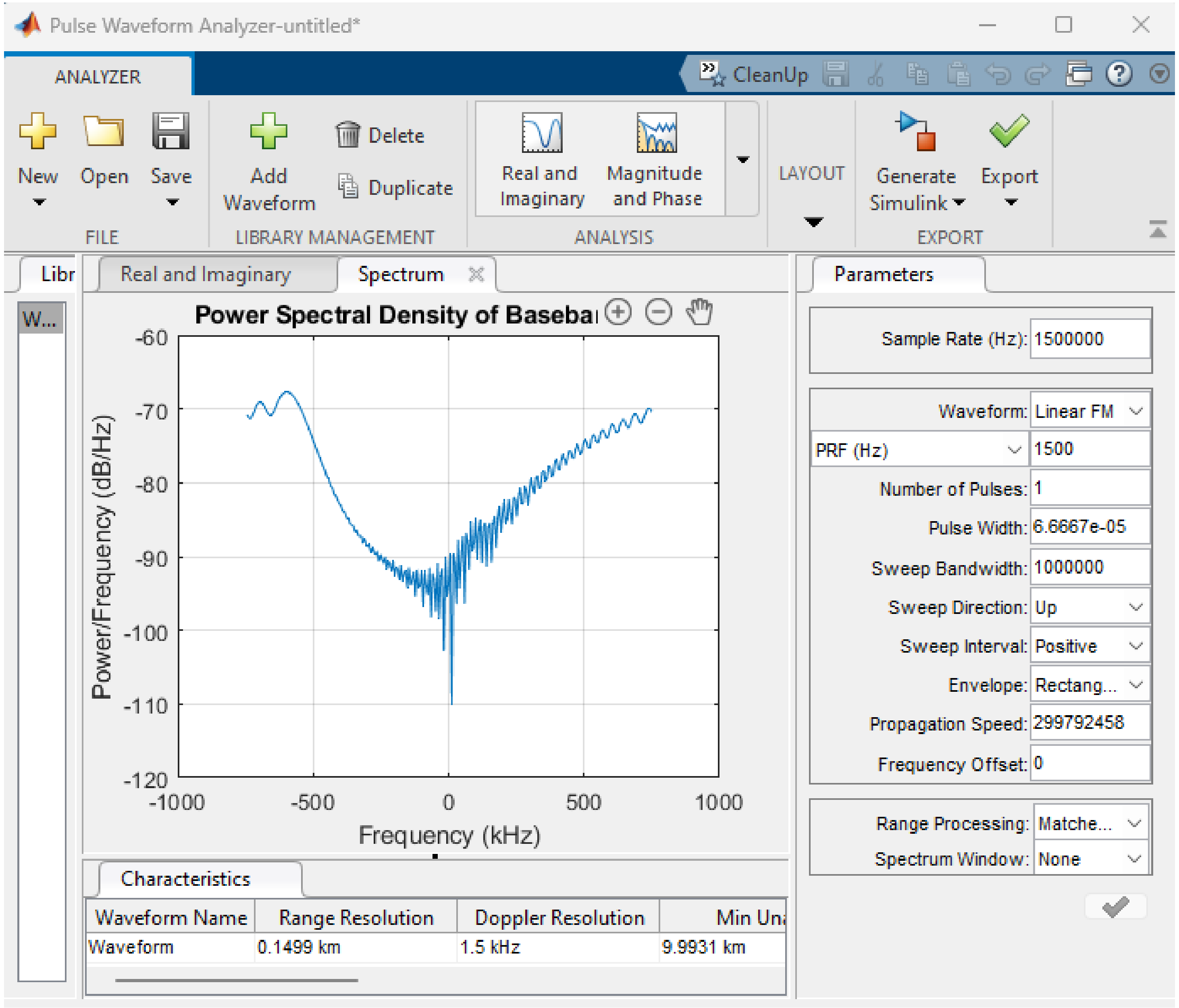
The waveform parameters are:

- Sample Rate = F_s = 1.5 MHz
- Pulse Repetition Frequency = PRF = 1.5 kHz
- Pulse Width = PW = 2/30k = 66.67 ms
- Sweep Bandwidth = BW = 1 MHz

Keep all Other parameters at Default

Exercise #1 *(Waveform – Con't)*

- Load the Pulse Waveform Analyzer App
- Choose Linear FM Waveform
- PRF = 1500 Hz
- Number of Pulses = 1
- Pulse Width = 66.667e-6
- Sweep Bandwidth = 1e6
- Sample Rate = 1.5e6
- Leave all other fields unchanged
- Click done (√)
- Export to MATLAB Script



Basic Radar Design Exercise # 1*(LFM Waveform export)*

```
% MATLAB Code from Radar Waveform Analyzer App
% Generated by MATLAB 9.14 and Phased Array System Toolbox 5.0
% Generated on 03-Oct-2023 11:28:13
% Create a Linear FM Waveform object
Waveform = phased.LinearFMWaveform('SampleRate',1.5e+06,...
'PulseWidth',6.6667e-05,'PRF',1500,'SweepBandwidth',1e+06,...
'SweepDirection','Up','SweepInterval','Positive'...
,'Envelope','Rectangular','NumPulses',1,'FrequencyOffset',0);
Fs = Waveform.SampleRate;

% Call step method of Waveform to generate IQ samples of Waveform
x = Waveform();

% Create figure for Real and Imaginary
figure;
l = (0:length(x)-1)/Fs;
subplot(2,1,1);
[~, scale, ~] = engunits(l(end));
l = l*scale;
plot(l,real(x));
axis([0 700 -1.0914 1.1]);
xlabel(''); ylabel('Amplitude: Real Part (V)'); grid on;
subplot(2,1,2);
plot(l,imag(x));
axis([0 700 -1.1 1.0998]);
xlabel('Time (\mus)');
ylabel('Amplitude: Imaginary Part (V)');
grid on;

% Create figure for Spectrum
figure;
pwelch(x, [], [], [], Fs, 'centered');
xlabel('Frequency (kHz)');
ylabel('Power/Frequency (dB/Hz)');
title('Power Spectral Density of Baseband Signal');

% Create a Matched Filter object
coeff = getMatchedFilter(Waveform);
Compression =
phased.MatchedFilter('Coefficients',coeff(:,1), ...
'SpectrumWindow','None');

y = Compression(x);
```

Exercise #2 *(Sensor Array Analyzer)*

- Use the Sensor Array Analyzer to design a 10x10 Array
- Export it as a MATLAB Script
- Insert it in the place indicated in Exercise1.m

Choose URA from Array Geometry Tab

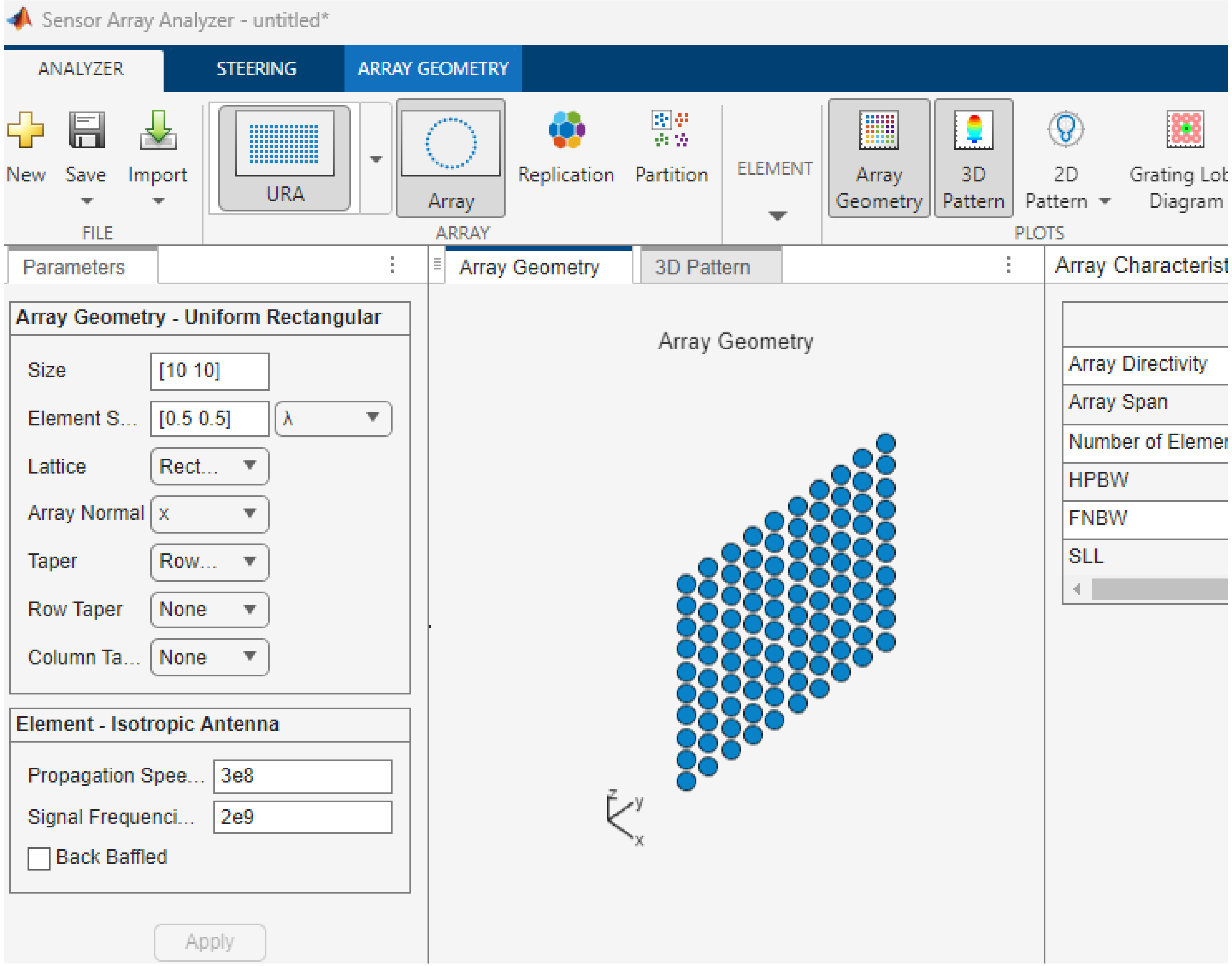
- Size = [10 10]
- Element Spacing = [0.5 0.5] λ
- Signal Frequency = 2e9 Hz

Keep all Other parameters at Default (including Isotropic Antenna Element)

Plot 3D directivity, and observe, Gain, and BeamWidth

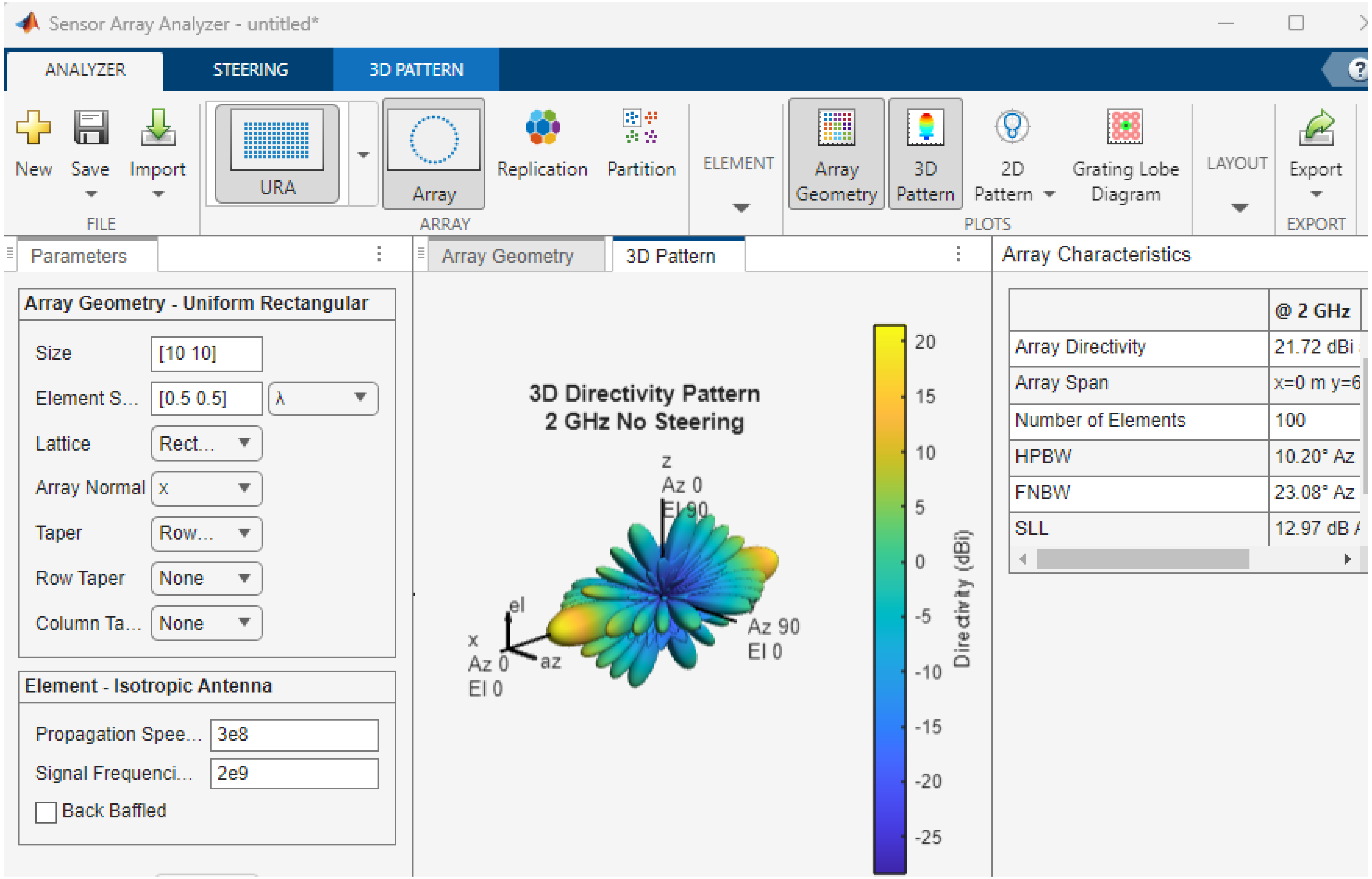
Basic Radar Design Exercise #2 *(Sensor Array –Con't)*

- Load the Pulse Waveform Analyzer App
- Choose URA and fill out the following parameters:
- Size = [10 10]
- Element Spacing = [0.5 0.5]λ
- Lattice = Rectangular
- Array Normal = x
- Taper = None (Row & Column)
- Element = Isotropic
- Signal Freq (Hz) = 2e9
- Click "Apply"
- Export to MATLAB Script



Basic Radar Design Exercise #2 *(Sensor Array –Con't)*

- Click 3D Pattern
- Export to MATLAB Script



Exercise #2 *(Sensor Array Export)*

```
% MATLAB Code from Sensor Array Analyzer App
% Generated by MATLAB 9.14 and Phased Array System Toolbox 5.0

% Create a uniform rectangular array
Array = phased.URA('Size',[10 10],...
'Lattice','Rectangular','ArrayNormal','x');
% The multiplication factor for lambda units to meter conversion
Array.ElementSpacing = [0.5 0.5]*0.15;
% Calculate Row taper
rwind = ones(1,10).';
% Calculate Column taper
cwind = ones(1,10).';
% Calculate taper
taper = rwind*cwind.';
Array.Taper = taper.';

% Create an isotropic antenna element
Elem = phased.IsotropicAntennaElement;
Elem.BackBaffled = true;
Elem.FrequencyRange = [0 2000000000];
Array.Element = Elem;

% Assign Frequencies and Propagation Speed
Frequency = 2000000000;
```

```
PropagationSpeed = 300000000;

% Create Figure
% Plot Array Geometry
figure;
viewArray(Array,'ShowNormal',false,...
'ShowTaper',false,'ShowIndex','None',...
'ShowLocalCoordinates',true,'ShowAnnotation',false,...
'Orientation',[0;0;0]);

% Calculate Steering Weights
Freq3D = 2000000000;
% Find the weights
w = ones(getNumElements(Array), length(Frequency));

% Plot 3d graph
format = 'polar';
plotType = 'Directivity';
figure;
pattern(Array, Freq3D , 'PropagationSpeed', PropagationSpeed,...
'CoordinateSystem', format,'weights', w(:,1),...
'ShowArray',false,'ShowLocalCoordinates',true,...
'ShowColorbar',true,'Orientation',[0;0;0],...
'Type', plotType);
```

MathWorks

Exercise #3 *(Radar Designer App)*

In this part we will use the Radar Designer App to find the parameters listed below given the specs introduced in the 1st few slides. *(for simplicity, we will only monitor P_d.)*

Load the Airport Radar Template of the Radar Designer App.

Find

- Peak Power
- Number of Pulses for Integration
- Tx Amp Gain
- Rx Amp Gain

Given that:

- f_c = 2 GHz
- Max Range = 100 km
- Min RCS = 0.1 m^2
- P_d = 0.9 , P_{fa} = 10^{-6}
- Waveform Param as in Exercise 1
- Array Param as in Exercise 2
- Tx Amp Gain = 8 to 12 dB
- Rx Amp Gain = 8 to 12 dB
- Rx Noise Figure = 7 dB
- Tx & Rx Amp Gain are equal
- Number of Pulses for Coherent Integration = 8 to 15 pulses

Basic Radar Design

Exercise #3 *(Radar Designer App – Con't)*

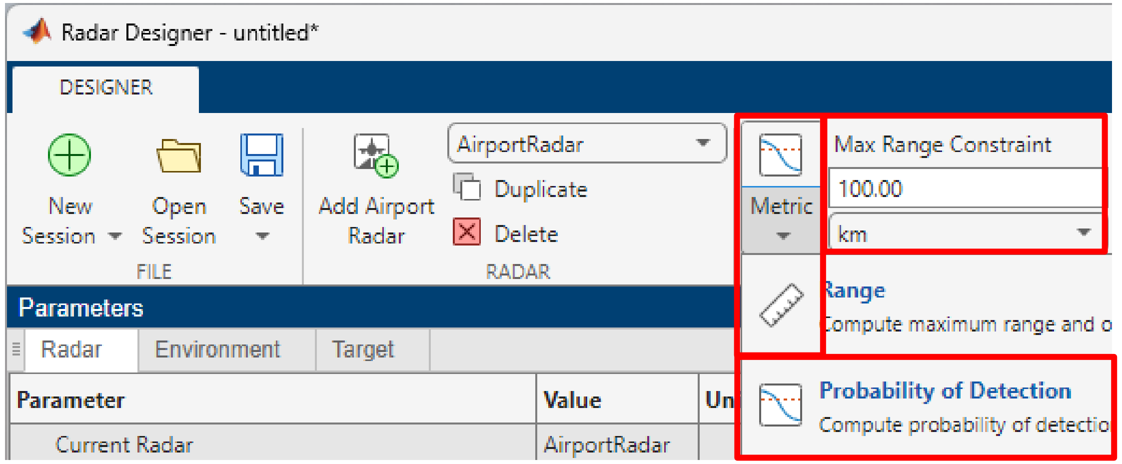
1) Set the Design to be constrained by the "Probability of Detection"

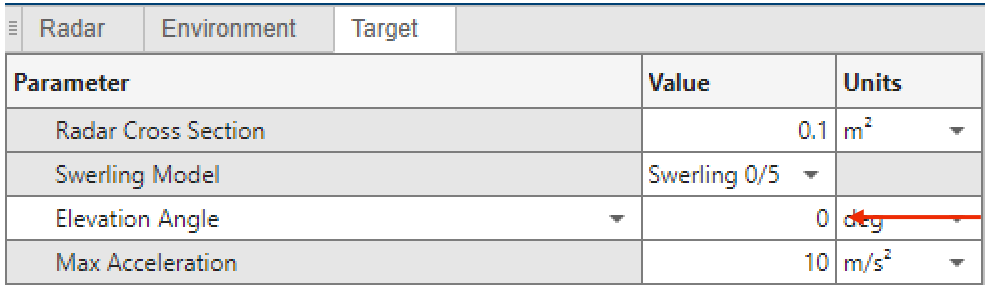
2) and the Maximum Range to 100 km

3) Set the Target Tab as shown

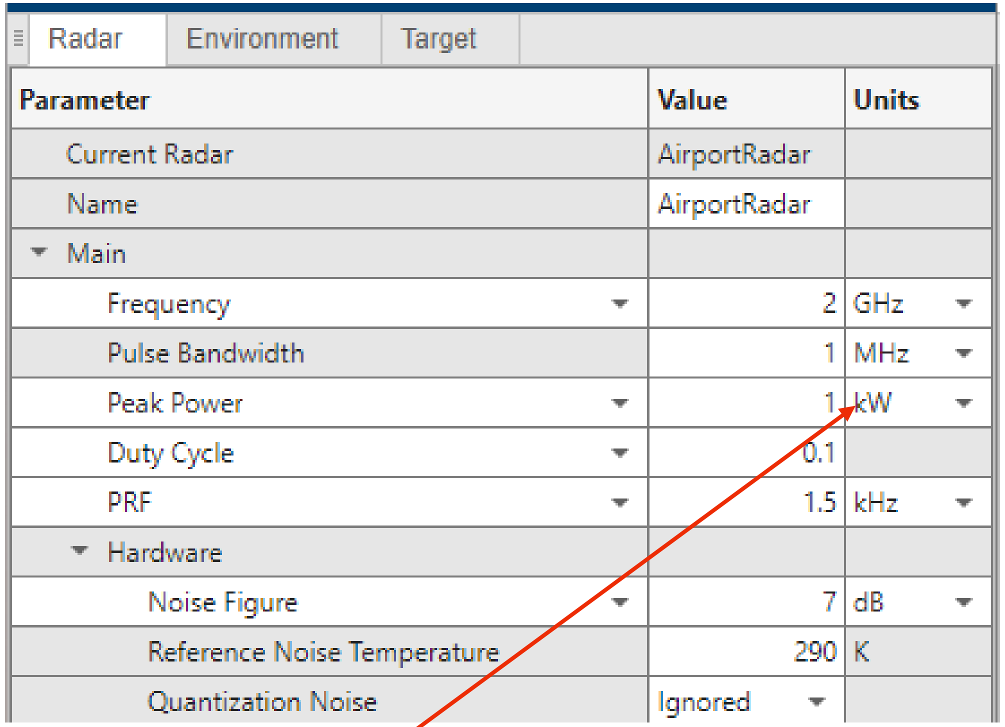
4) Leave the Environment Tab at Default *(Free Space and No precipitation)*

5) Fill in the "Main" parameters in the Radar Tab with design parameters.





Same Level as Radar



Start with 100 W

MathWorks

Exercise #3 *(Radar Designer App Con't)*

- Fill in the "Antenna and Scanning" parameters with the design parameters.

Radar | Environment | Target

Parameter	Value	Units
Current Radar	AirportRadar	
Name	AirportRadar	
▸ Main		
▾ Antenna and Scanning		
Antenna Height	100	m
Antenna Tilt Angle	0	deg
Antenna Polarization	Horizontal	
▾ Transmit Antenna Gain Input	Manual	
Gain	21.73	dBi
Use Different Antenna for Receive	False	
Scan Mode	None	
▸ Detection and Tracking		
▸ Loss Factors		

Irrelevant

Same Level as Target

Obtained from Exercise 2

Fixed Beam – No Scanning

- Fill in the detection and Tracking parameters in the Radar Tab
- Do the same for the Loss Factors.

Radar | Environment | Target

Parameter	Value	Units
Current Radar	AirportRadar	
Name	AirportRadar	
▸ Main		
▸ Antenna and Scanning		
▾ Detection and Tracking		
Probability of False Alarm	0.000001	
Number of Pulses	12	
Pulse Integration	Coherent	
Constant False Alarm Rate (CFAR)	Unspecified	
Number of CPIs	1	
M-of-N CPI Integration	Off	
Sensitivity Time Control (STC)	Off	
▸ Track Confirmation Logic		
▾ Loss Factors		
Eclipsing	None	
Custom Loss	-28.24	dB

Start with 12

Negative Loss is Amplification

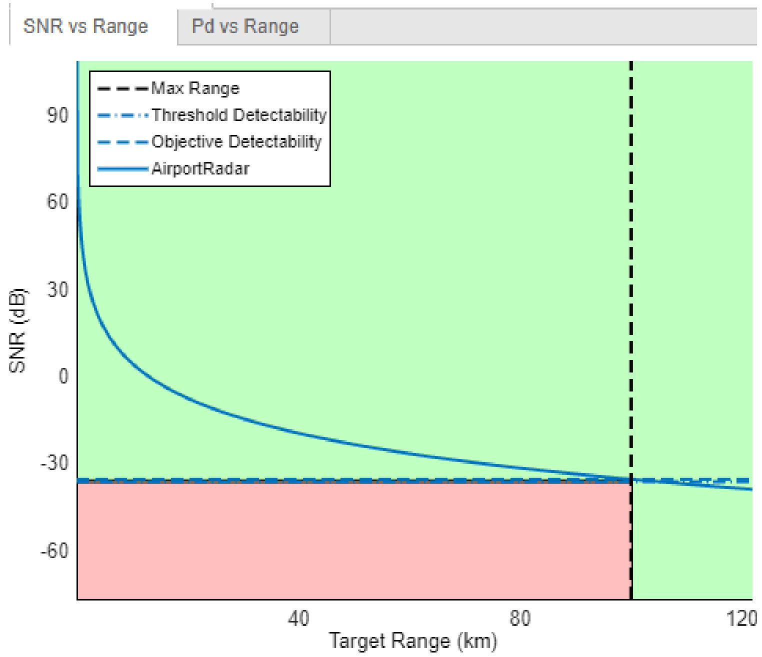
Tx_Gain + Rx_Gain + MatchFilt_Gain

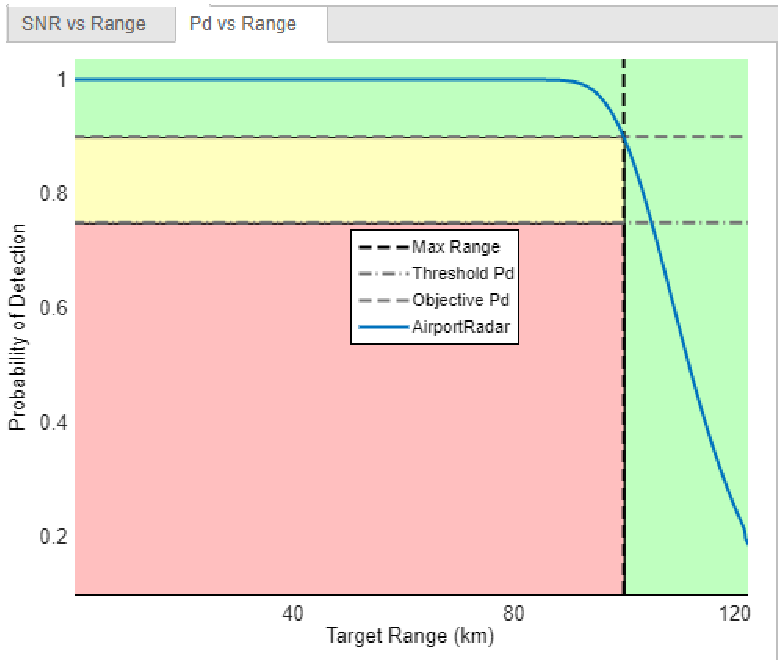
Start with – 38.24 dB

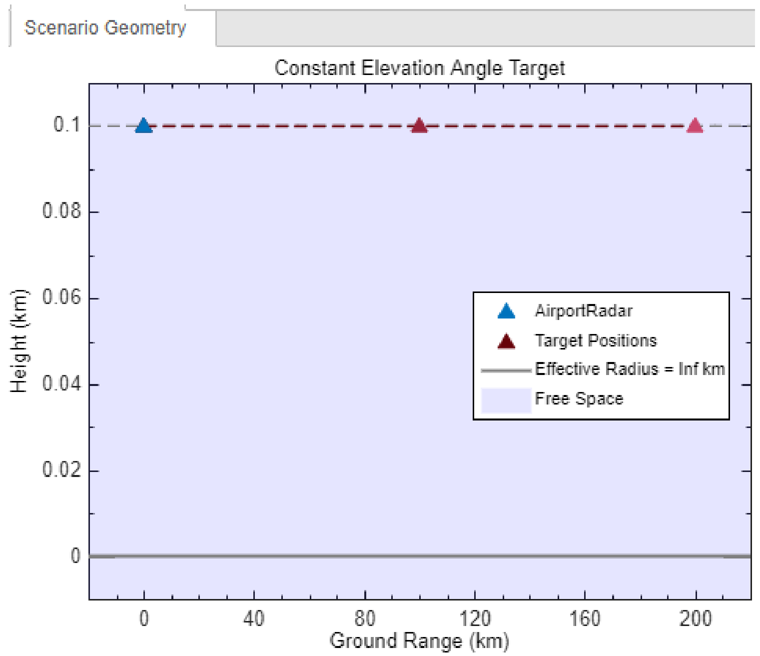
Exercise #3 *(Radar Designer App Con't)*

Although there is more than 1 set of values that meet the specifications, we will use the following for Simulation and Detection in Exercise 4

- Ppower = 369 W
- # of Pulses = 10
- Tx_Amp_Gain = 10 dB
- Rx_Amp_Gain = 10 dB







Exercise #4 *(Simulation & Detection)*

In Exercise 4 we will use the results of Exercises 1, 2 and 3 , to set up Simulation & Detection

- Set up Linear FM Waveform (Done in Exercise 1)
- Radar Antenna Array (Done in Exercise 2)

Complete the missing parameters in:

- Transmitter, Receiver, and radarTranceiver System Object (All done)
  - Gain, and Peak Power
- Target, Scene, and Environment.
  - Position, Velocity and RCS
- Signal Processing: Target Detection
  - Noise Floor , and Detection Threshold

Run and observe

- Noise Floor of Plot as compared to calculated Noise Floor
- Level of Detected signal at 90 km to the specified P_d and P_f.

What are some next options?

Multi-Function Radar Workshop

If this workshop as useful to you, you might want to consider parts 2&3

- Part 1: Basic Radar Design
- **Part 2: Scanning Radar**
- **Part 3: Search & Track Radar**

Abstract for Complete Workshop

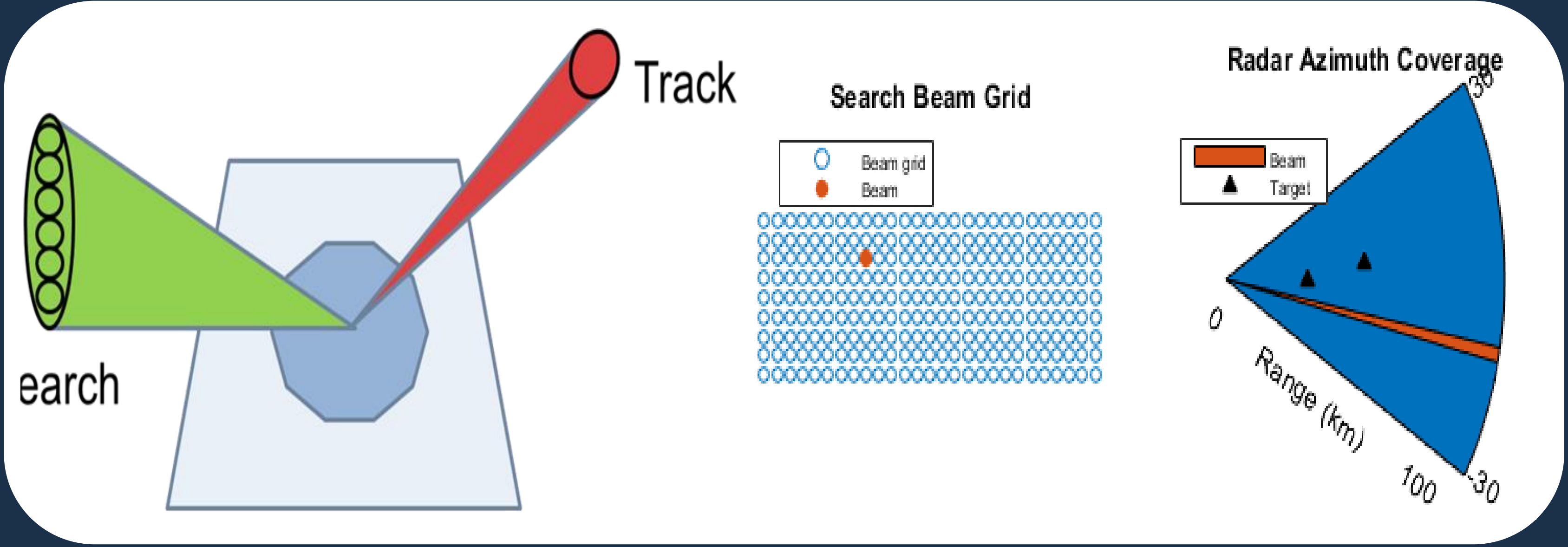
A multifunction radar can perform jobs that usually require multiple traditional radars. Examples of such Radars are:

- Traditional radars are scanning radars, which are responsible for searching targets
- Tracking radars, which are responsible for tracking targets.

In this example, the multifunction phased array radar performs both scanning (searching) and tracking tasks. Based on the detections and tracks obtained from the current echo, the radar decides what to do next to ensure that targets of interest are tracked and the desired airspace is searched.

Multi-Function Radar Workshop

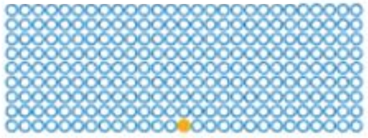
Resource Management & Controls



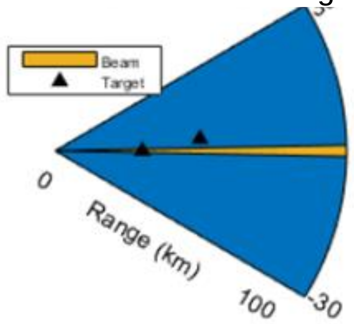
Multi-Function Radar Workshop

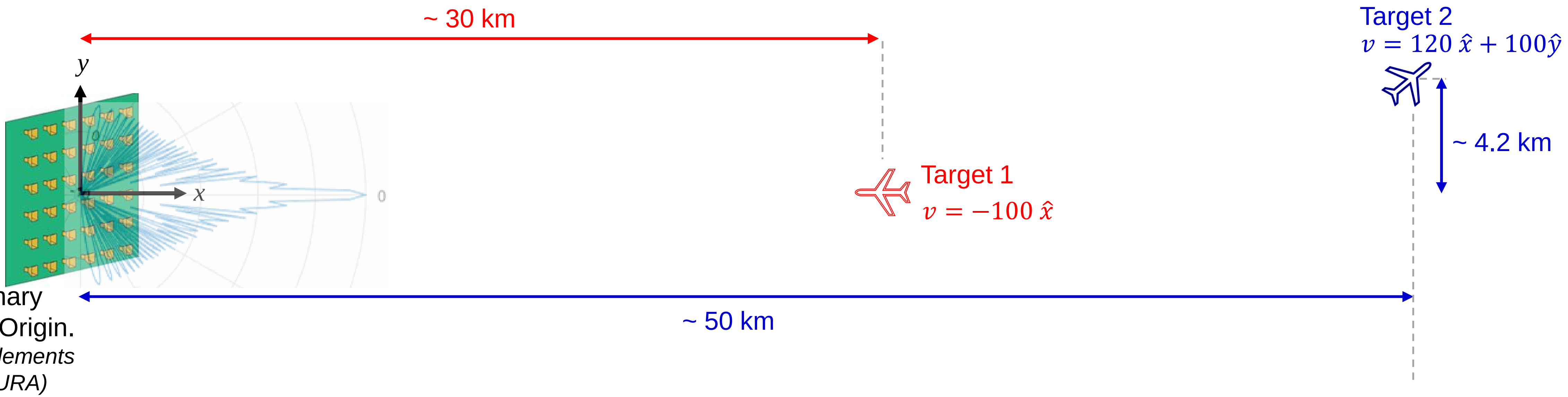
Scenario Overview

Search Beam Grid
Az = -30 to 30
El = 0 to 20



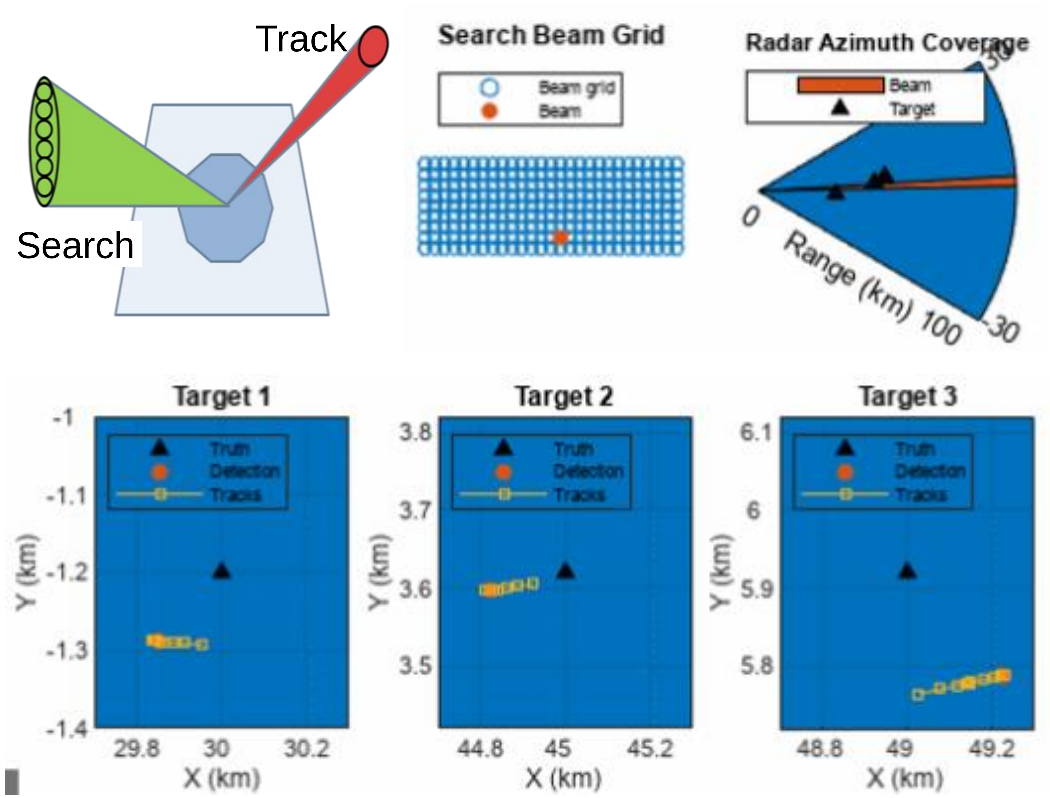
Radar Azimuth Coverage





Stationary
Radar at Origin.
Isotropic Elements
(10x10 URA)

Final Result



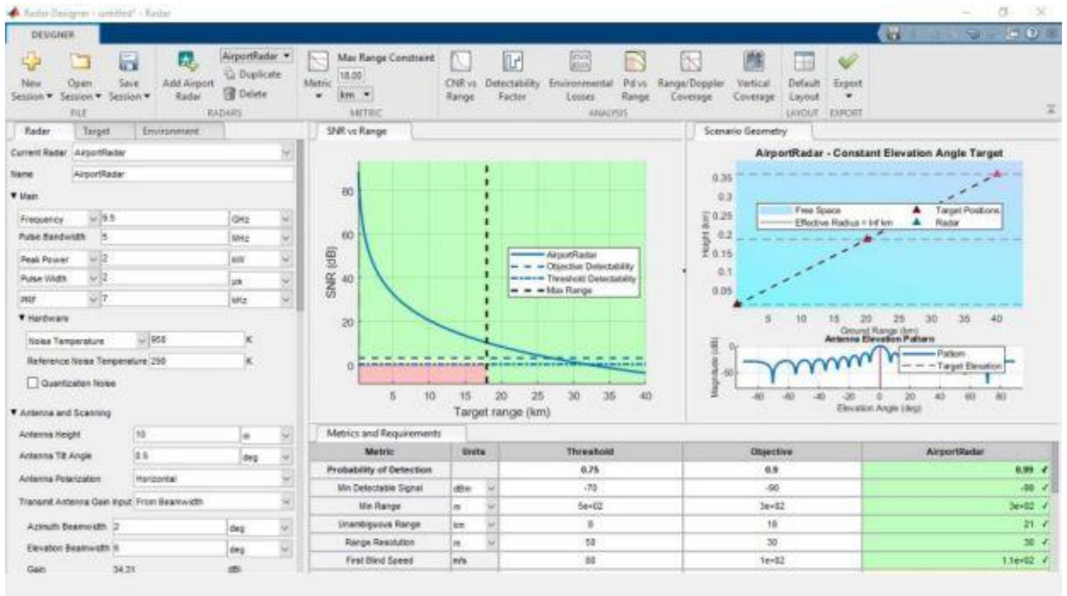
Full Instructor-led Courses (1+ days)

Modeling Radar Systems with MATLAB

***Available Private Only**

- Radar systems engineering for preliminary radar design
- Scenario generation including targets, propagation, and terrain
- System simulation at the measurement and physics (IQ signal) level
- Signal processing for target detection
- Data processing for multi-object tracking
- Higher fidelity modeling including antenna array design, waveform analysis, clutter, polarization, and micro-Doppler signature generation
- Radar data synthesis and labeling for deep learning

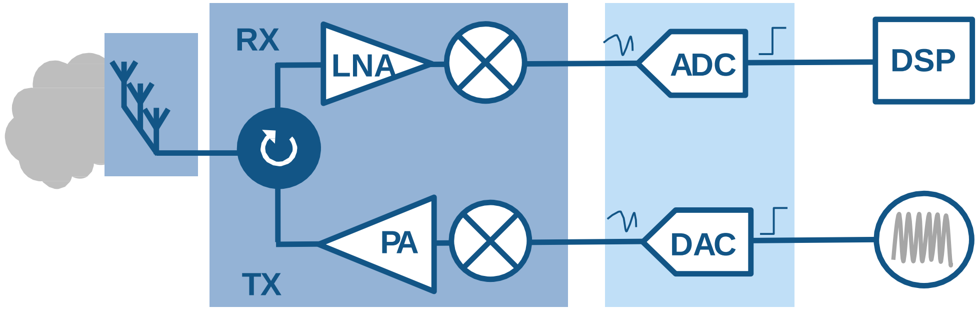
[See detailed course outline]

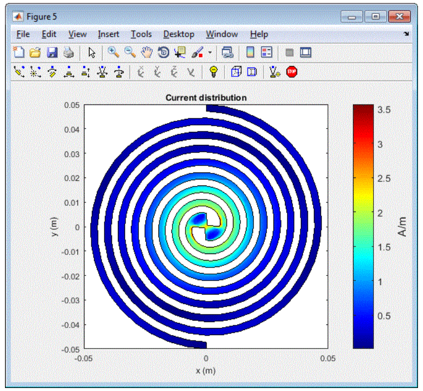


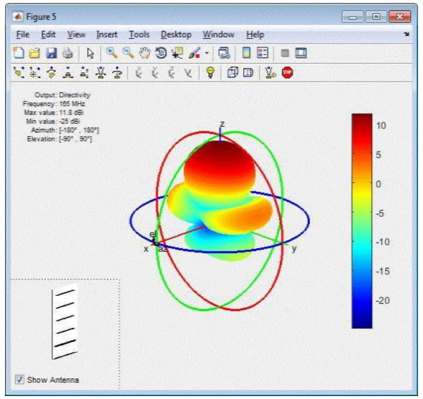
Modeling Wireless Communications Systems using Phased Array System Toolbox

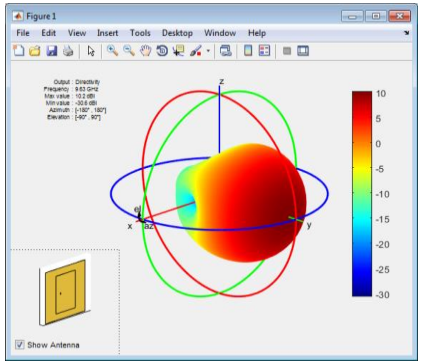
Topic included in this 1-day training:

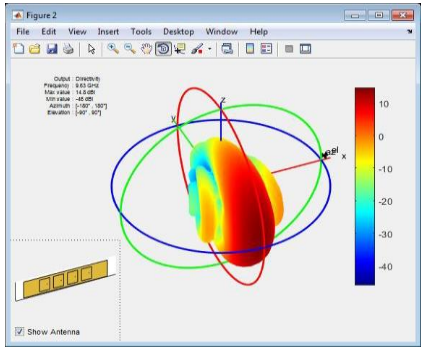
- Phased Array Design and Analysis
- Antenna Patterns and Mutual Coupling
- Design and Modeling Components of a Wireless System
- Spatial Signal Processing Algorithms









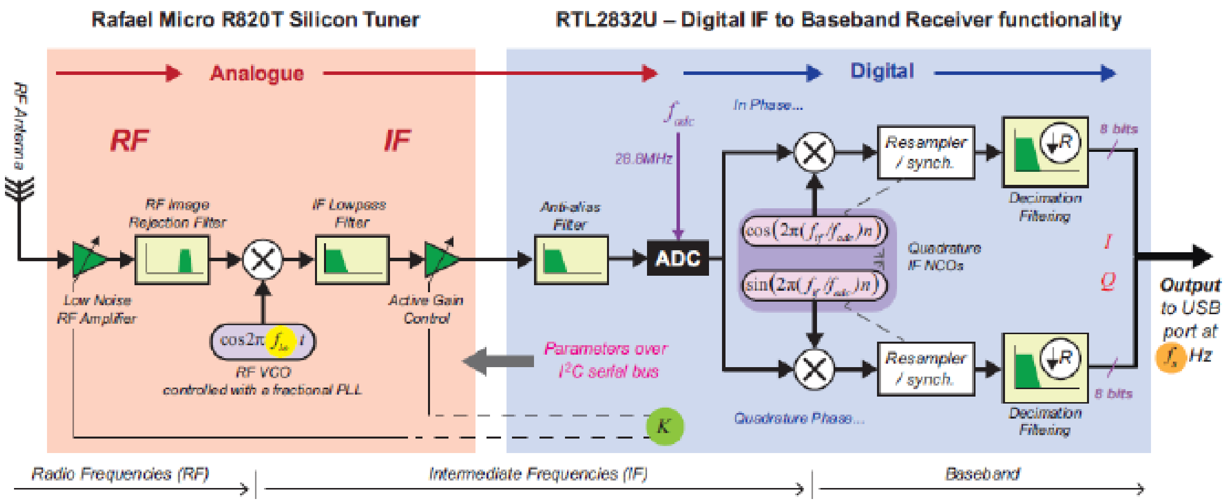


Note: this training is available only for private training events

Wireless Communications Systems Design with MATLAB and USRP Software-Defined Radios

Topics included in this 2-day course:

- Single- and multi-carrier communication
- Different channel impairment models
- Multiple antenna and Turbo code
- LTE and IEEE 802.11 example
- Radio-in-the-loop system



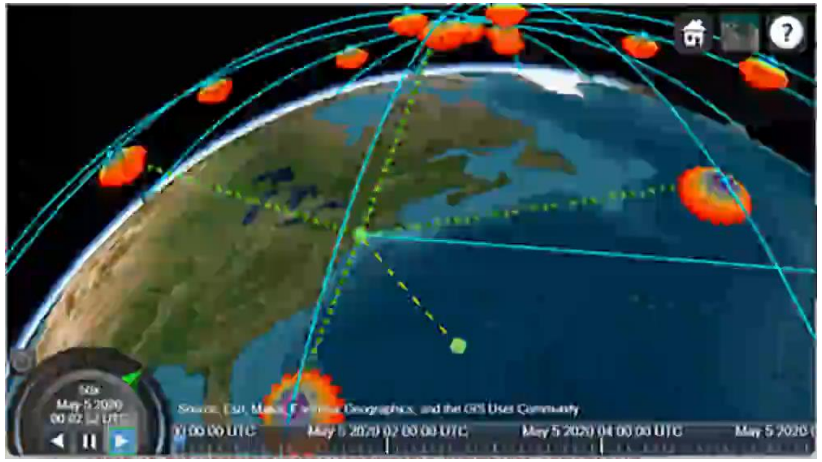
[See detailed course outline]

More Workshops (2-3 hours, free)

Satellite Communications Hands-On Workshop

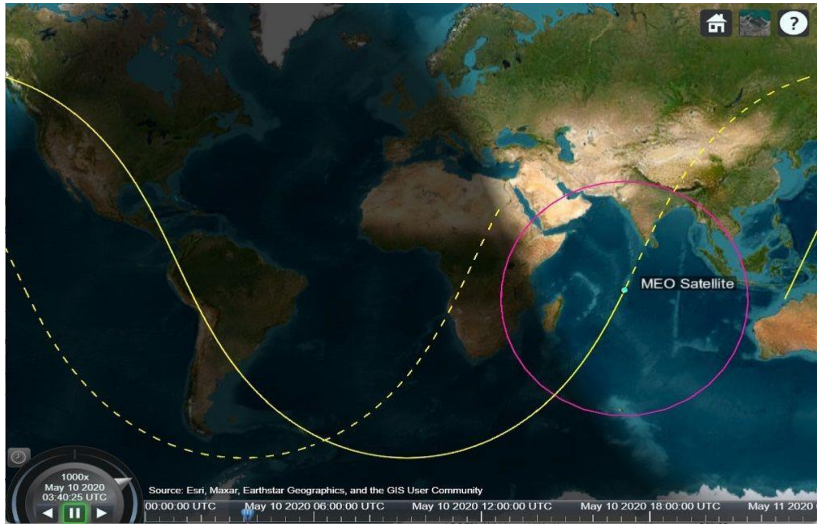
Overview

In this private hands-on workshop, MathWorks product experts will walk you through a series of online exercises. These guided exercises will give you the opportunity to write and run your own code using Satellite Communications Toolbox and learn how, with minimal coding, you can use the toolbox to streamline your satellite-related workflows.



Agenda

- Brief overview of Satellite Communications Toolbox
- Hands-on exercises using MATLAB Online where you will:
  - Set up and launch a satellite scenario viewer
  - Compute and visualize the visibility access between a satellite and a ground station
  - Compute and visualize communications link closure between a satellite and a ground station



Scheduled to run for 2 hours

AI for Wireless Applications

A Hands-On Workshop

Overview

Exercise 1: Getting Started with Deep Learning

Exercise 2: Building Networks from Scratch/ Modulation Classification

Exercise 3: Deep Learning Data Synthesis for 5G Channel Estimation

Scheduled to run for 3 hours

MathWorks®