MathWorks®

Hands-on Workshop :

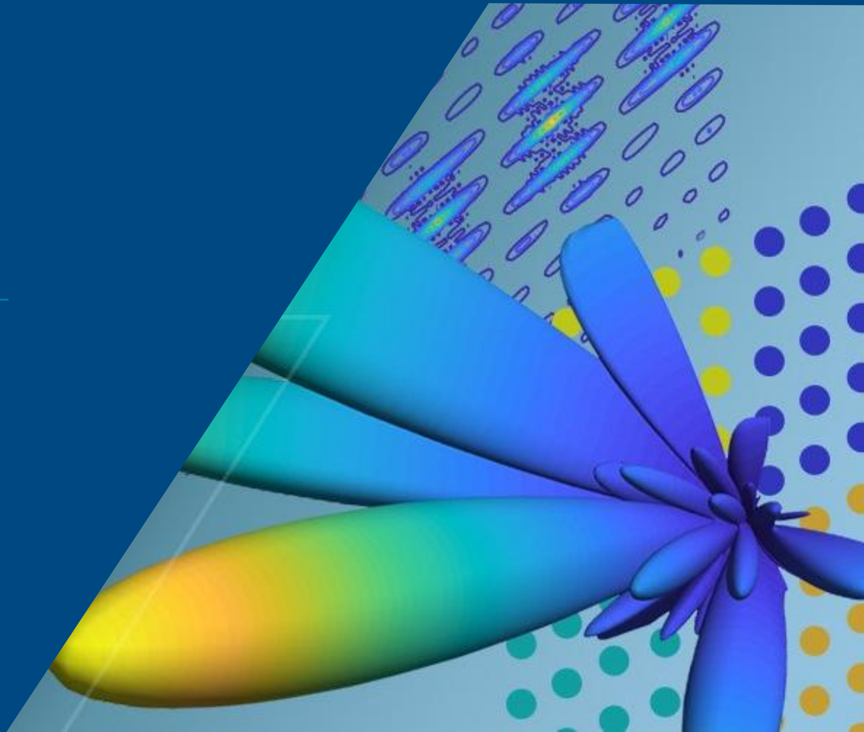
Phased Array Design and Simulation in MATLAB

Akash Gopisetty , Product Marketing

Tony Azar, Application Engineer

MathWorks

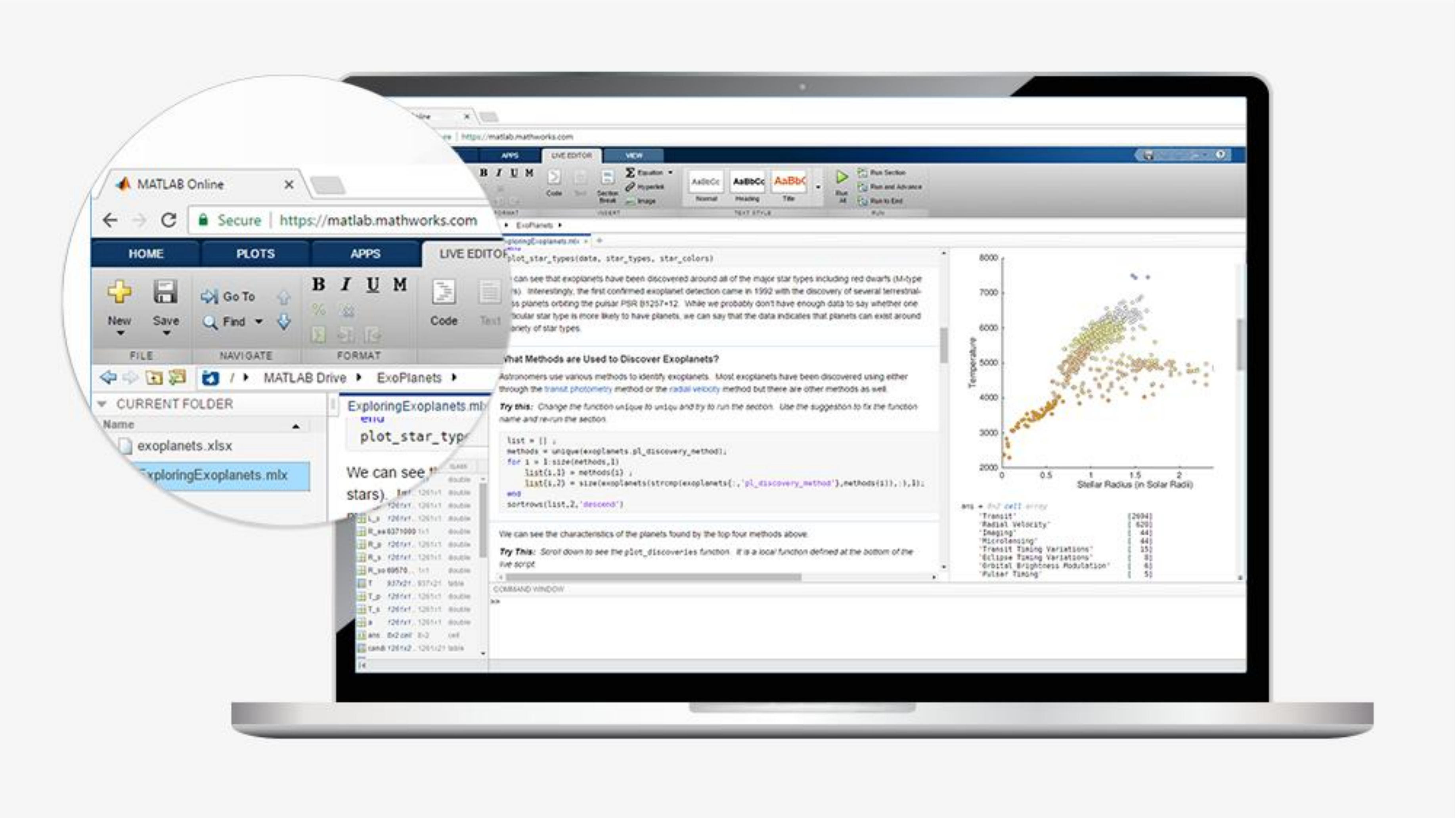
Dec 18, 2025



Agenda

1. Setup and Logistics
2. Introduction
3. Hands-on exercises
4. Takeaways and Resources

Use MATLAB Online for this workshop



```
plot_star_types(data, star_types, star_colors)
```

can see that exoplanets have been discovered around all of the major star types including red dwarfs (M-type s). Interestingly, the first confirmed exoplanet detection came in 1992 with the discovery of several terrestrial-ss planets orbiting the pulsar PSR B1257+12. While we probably don't have enough data to say whether one ticular star type is more likely to have planets, we can say that the data indicates that planets can exist around ariety of star types.

hat Methods are Used to Discover Exoplanets?

Astronomers use various methods to identify exoplanets. Most exoplanets have been discovered using either through the transit photometry method or the radial velocity method but there are other methods as well.

Try this: *Change the function* unique *to* uniqu *and try to run the section. Use the suggestion to fix the function name and re-run the section.*

```
list = [] ;
methods = unique(exoplanets.pl_discovery_method);
for i = 1:size(methods,1)
    list(i,1) = methods(i) ;
    list(i,2) = size(exoplanets(strcmp(exoplanets{:,'pl_discovery_method'},methods(i)),:),1);
end
sortrows(list,2,'descend')
```

```
ans = 8×2 cell array
    'Transit'                             [2694]
    'Radial Velocity'                     [ 620]
    'Imaging'                             [  44]
    'Microlensing'                        [  44]
    'Transit Timing Variations'           [  15]
    'Eclipse Timing Variations'           [   8]
    'Orbital Brightness Modulation'       [   6]
    'Pulsar Timing'                       [   5]
```

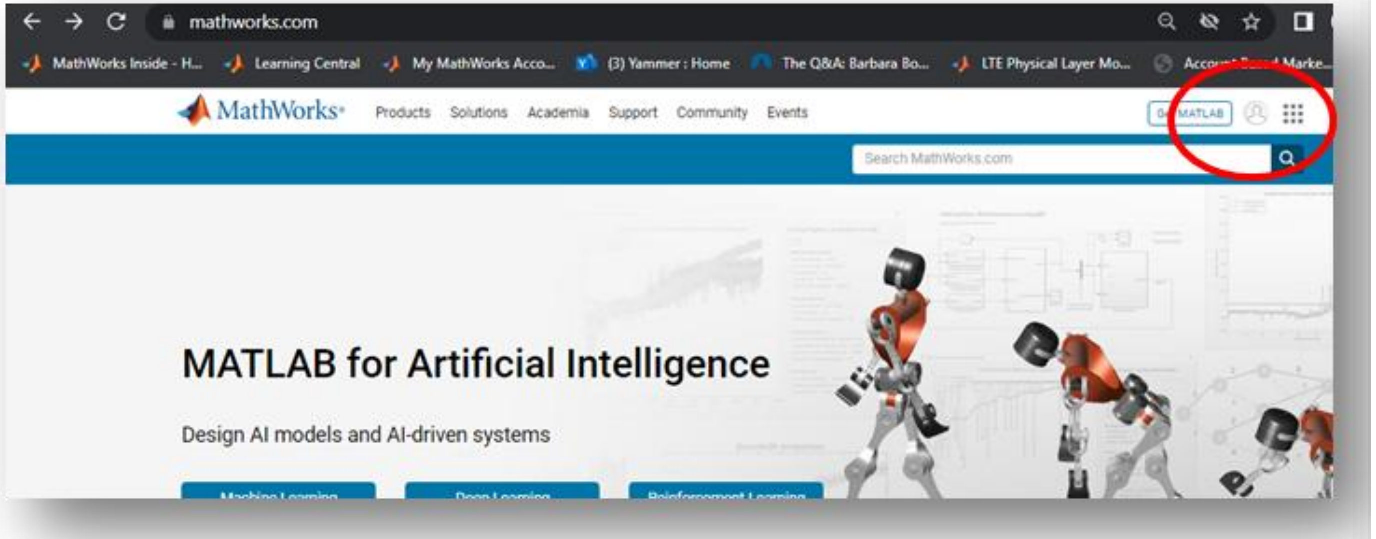
We can see the characteristics of the planets found by the top four methods above.

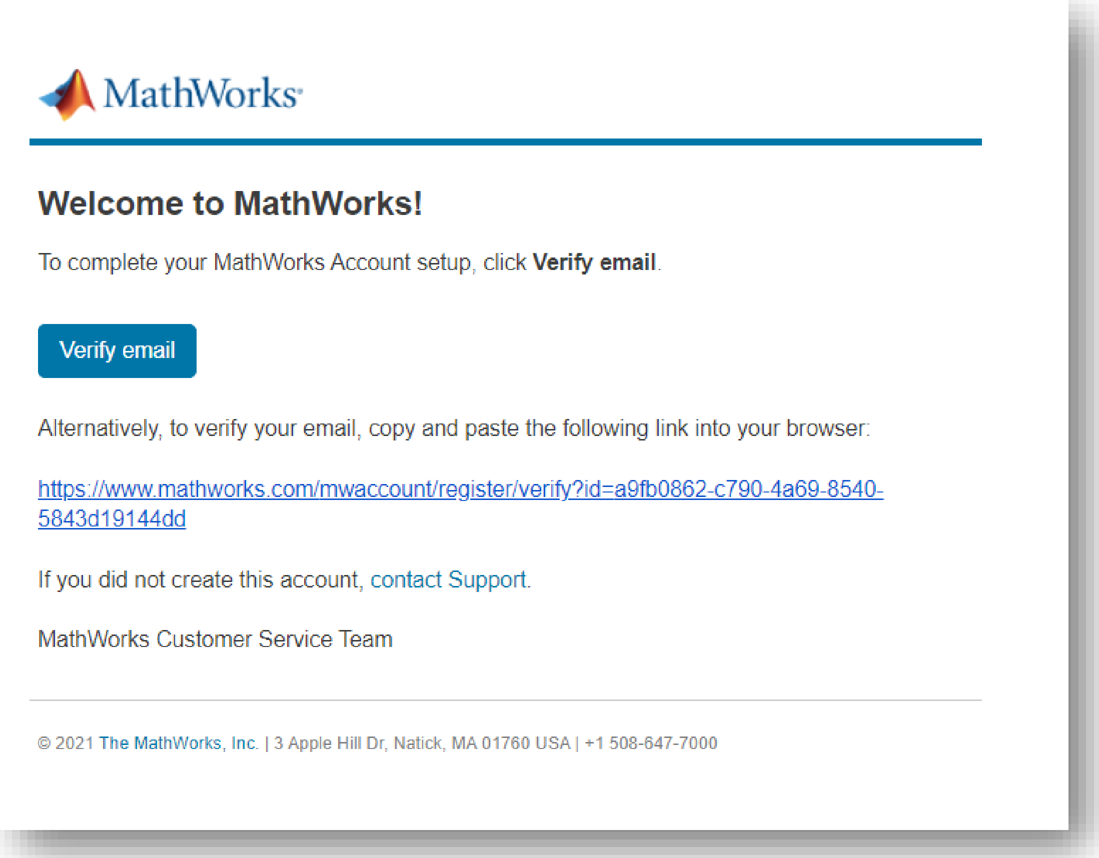
Try This: *Scroll down to see the* plot_discoveries *function. It is a local function defined at the bottom of the live script.*

```
>>
```

If you don’t already have MATLAB Online access:

- Go to www.mathworks.com and click on the profile image at the top right
- You will be prompted to create an account (please use your work address)
- Once you get your confirmation email, please verify your account





MathWorks

Welcome to MathWorks!

To complete your MathWorks Account setup, click **Verify email**.

Verify email

Alternatively, to verify your email, copy and paste the following link into your browser:

https://www.mathworks.com/mwaccount/register/verify?id=a9fb0862-c790-4a69-8540-5843d19144dd

If you did not create this account, contact Support.

MathWorks Customer Service Team

© 2021 The MathWorks, Inc. | 3 Apple Hill Dr, Natick, MA 01760 USA | +1 508-647-7000

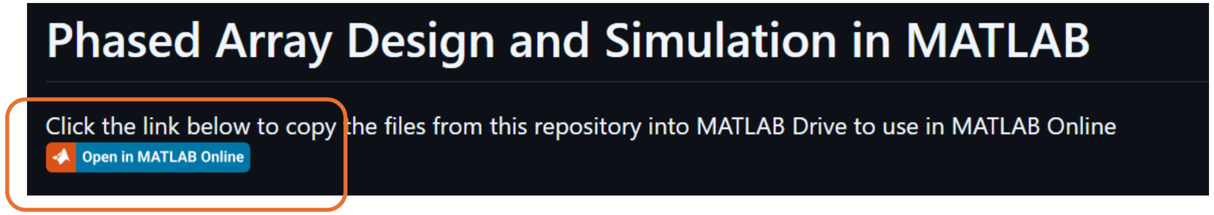
You will need two links for the workshop

1. Access MATLAB Online for 30 days:

   URL in your workshop correspondence

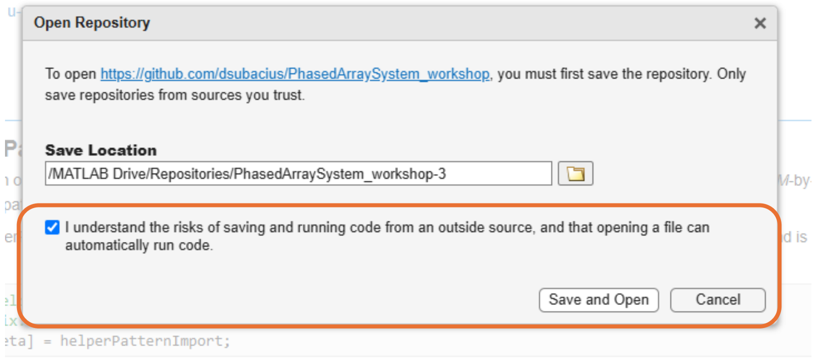
2. Workshop code files on GitHub:

   https://github.com/dsubacius/PhasedArraySystem_workshop/tree/main



Phased Array Design and Simulation in MATLAB

Click the link below to copy the files from this repository into MATLAB Drive to use in MATLAB Online



Open Repository

To open https://github.com/dsubacius/PhasedArraySystem_workshop, you must first save the repository. Only save repositories from sources you trust.

Save Location

/MATLAB Drive/Repositories/PhasedArraySystem_workshop-3

☑ I understand the risks of saving and running code from an outside source, and that opening a file can automatically run code.

Typical applications of phased arrays

Multifunction Radars

Wireless Communications

Satellite Communications

Audio / Underwater Acoustics

Design, Simulate and Analyze Phased Arrays

Array Design

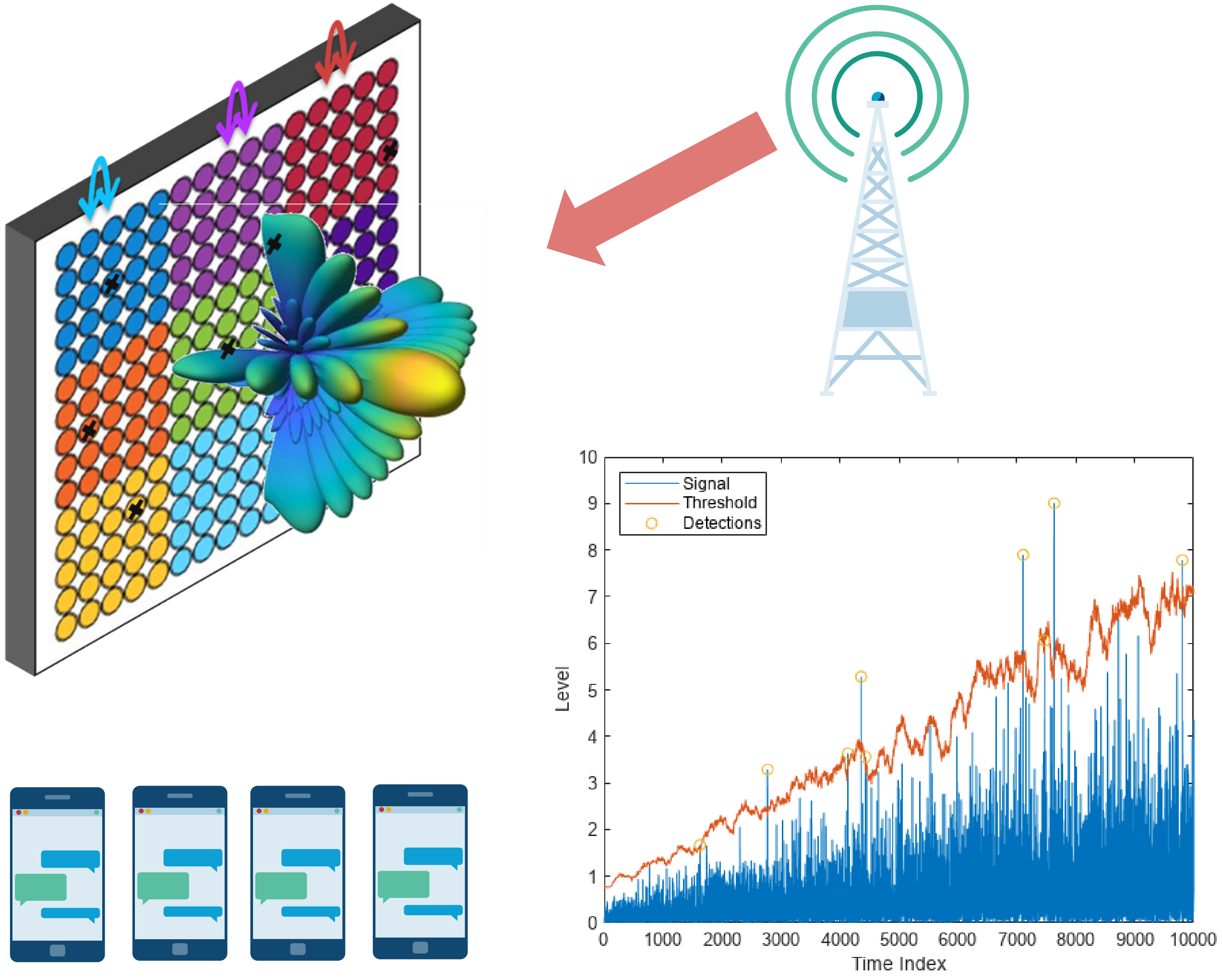
Mutual Coupling

Signal Processing

Interference Mitigation

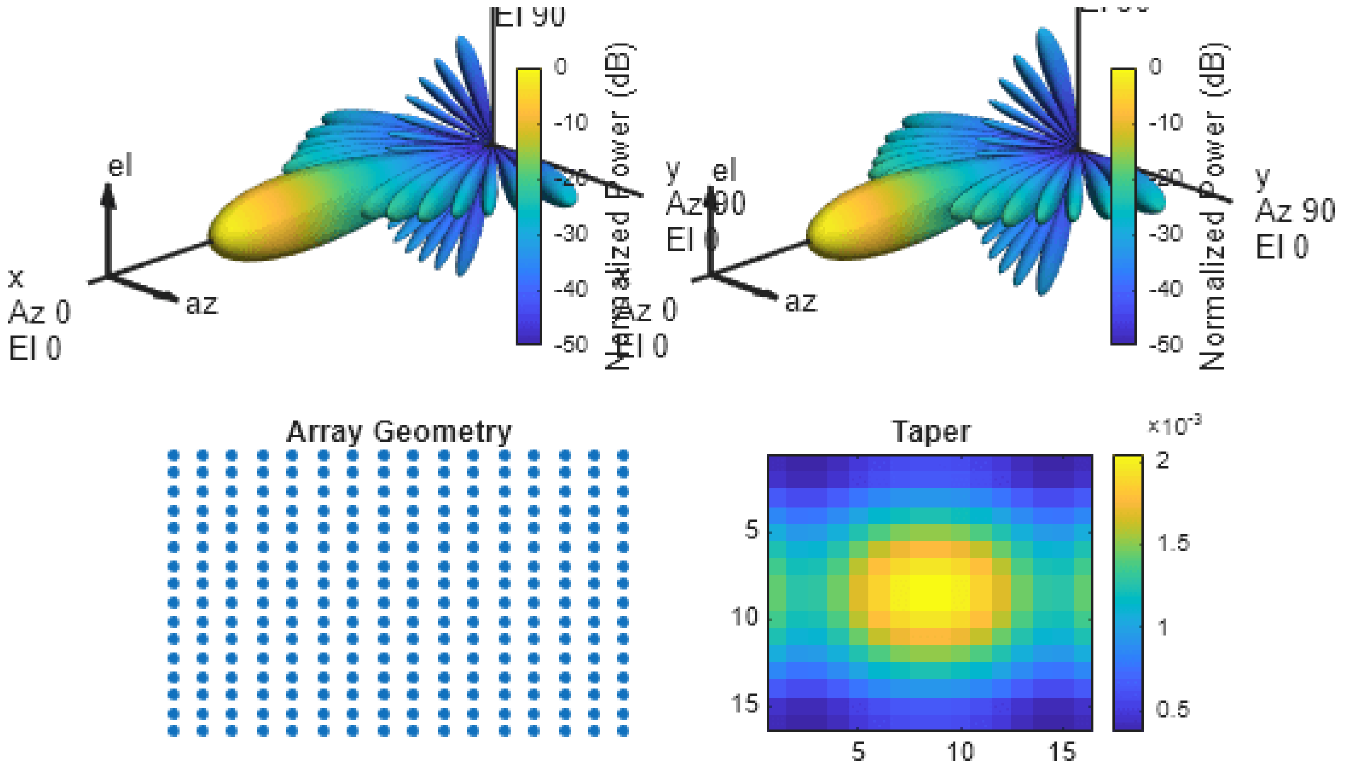
Calibration

Waveform Design

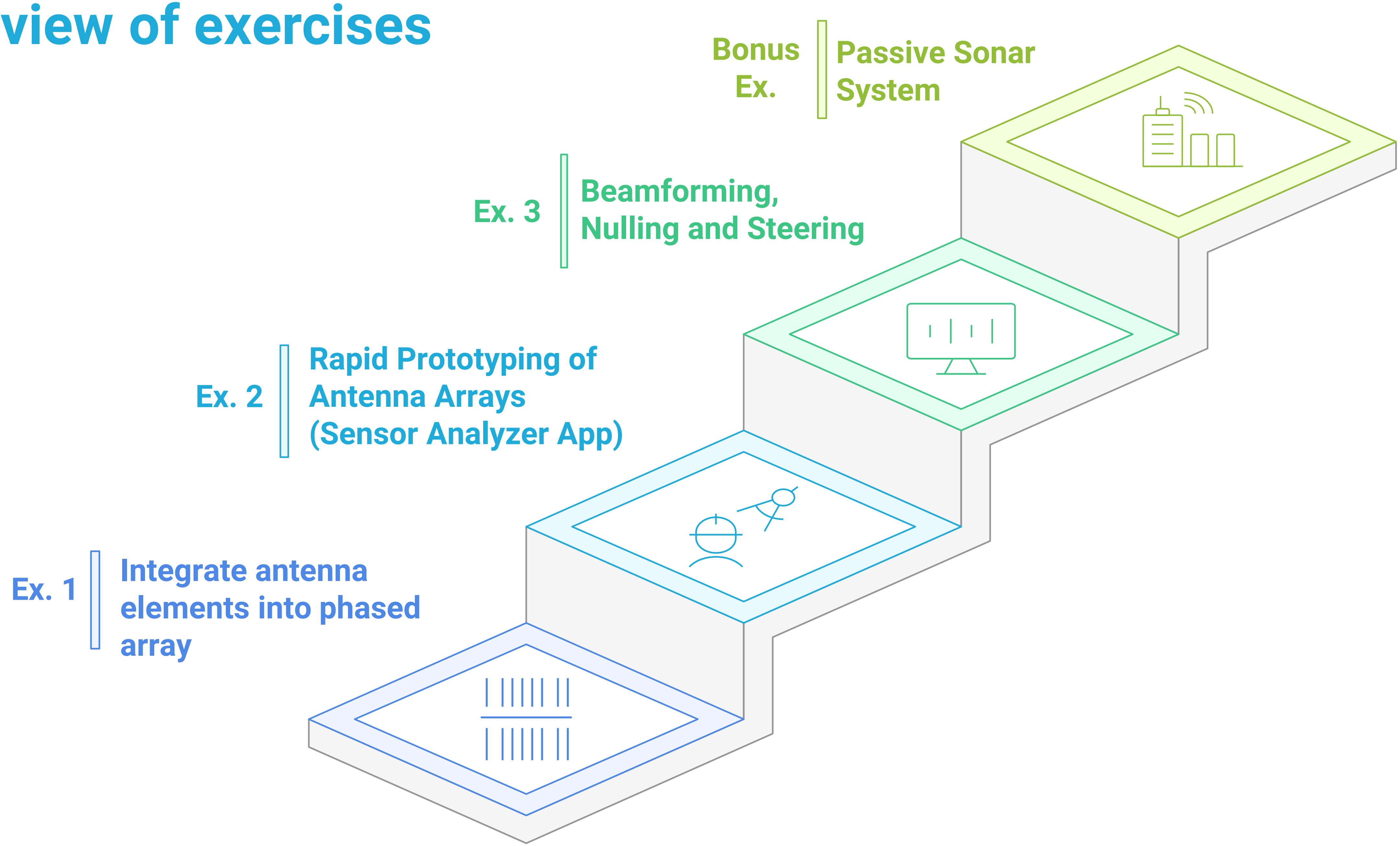


By the end of the workshop, you will be able to

- Get started with phased arrays in MATLAB and apply them in your projects
- Design and analyze various array configurations
- Understand the basics of pattern synthesis
- Integrate phased arrays into larger system-level models (sonar example)



Overview of exercises



Bonus Ex. | Passive Sonar System

Ex. 3 | Beamforming, Nulling and Steering

Ex. 2 | Rapid Prototyping of Antenna Arrays (Sensor Analyzer App)

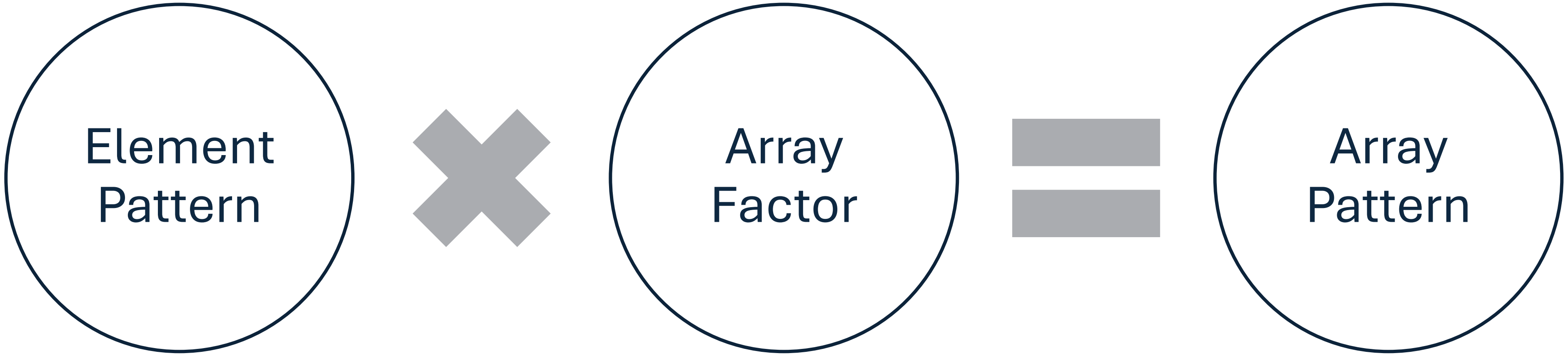
Ex. 1 | Integrate antenna elements into phased array

1

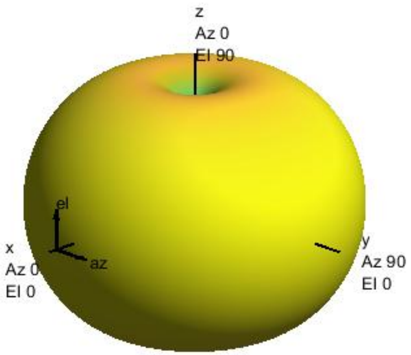
Integrate antenna elements into phased array

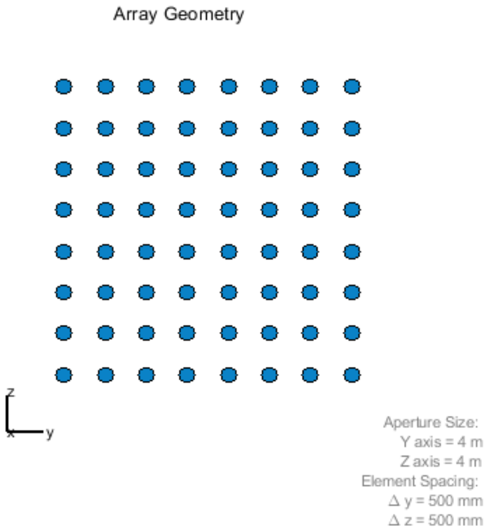
- Familiarize with Phased Array System toolbox objects using command line interface
- Learn how to import a custom antenna elements from other tools e.g., HFSS
- Construct and analyze a Uniform Rectangular Array (URA) using the defined antenna elements.
- Familiarize yourself with the MATLAB Online environment

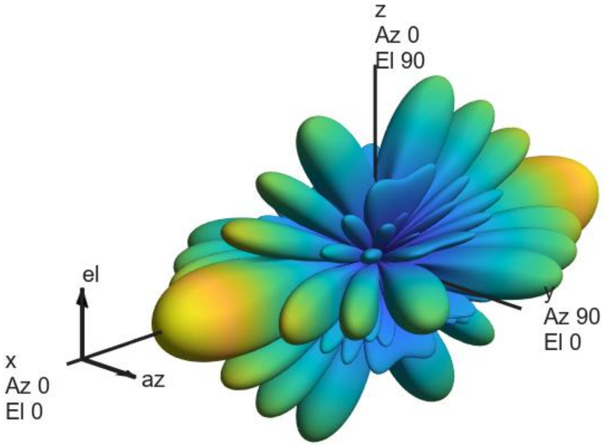
Array Patterns are Determined by Element Pattern and Spatial Configuration



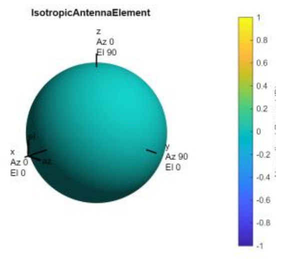
Element Pattern × Array Factor = Array Pattern



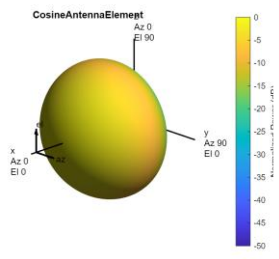




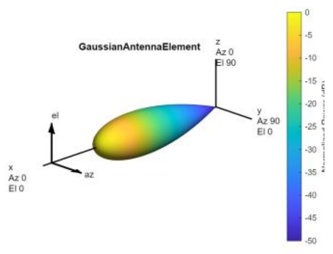
Options for element patterns



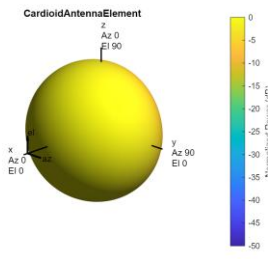
Isotropic Antenna



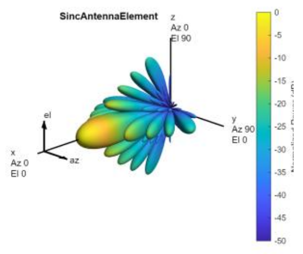
Cosine Antenna



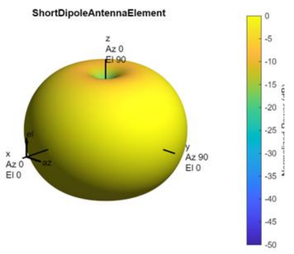
Gaussian Antenna



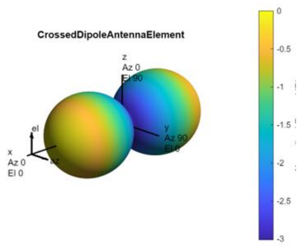
Cardioid Antenna



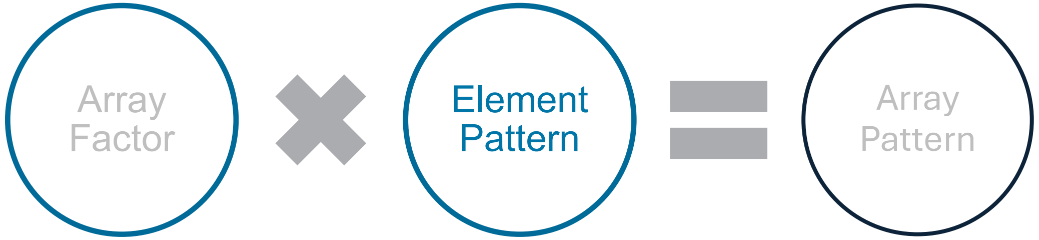
Sinc Antenna



Short Dipole



Crossed Dipole



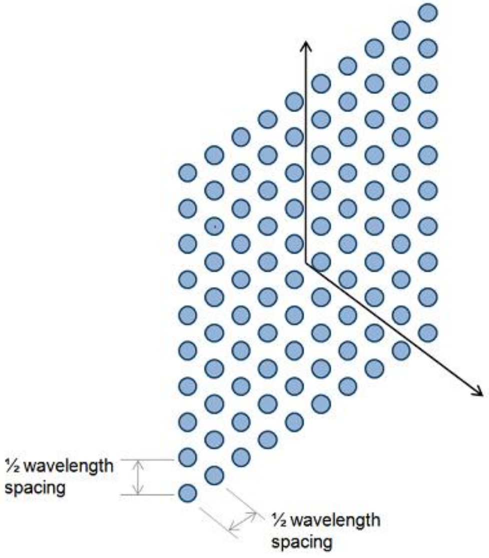
Array Factor

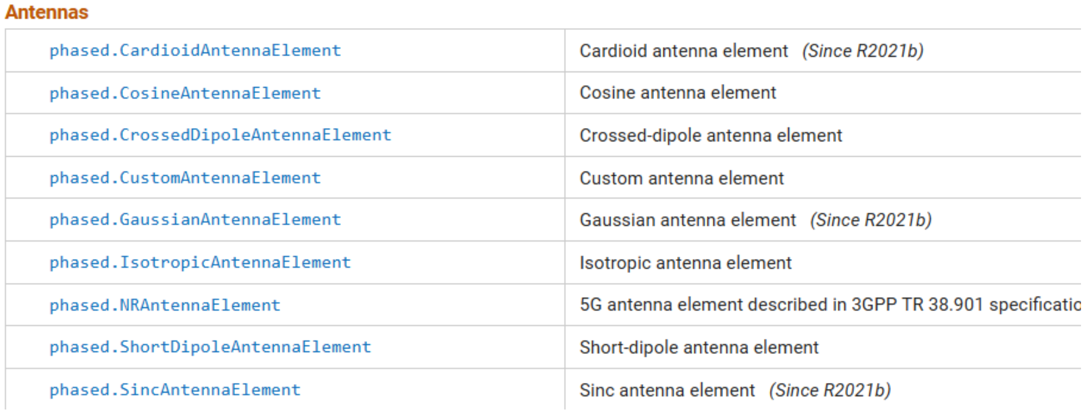
Element Pattern

Array Pattern

Phased Array System Toolbox uses functions and objects:

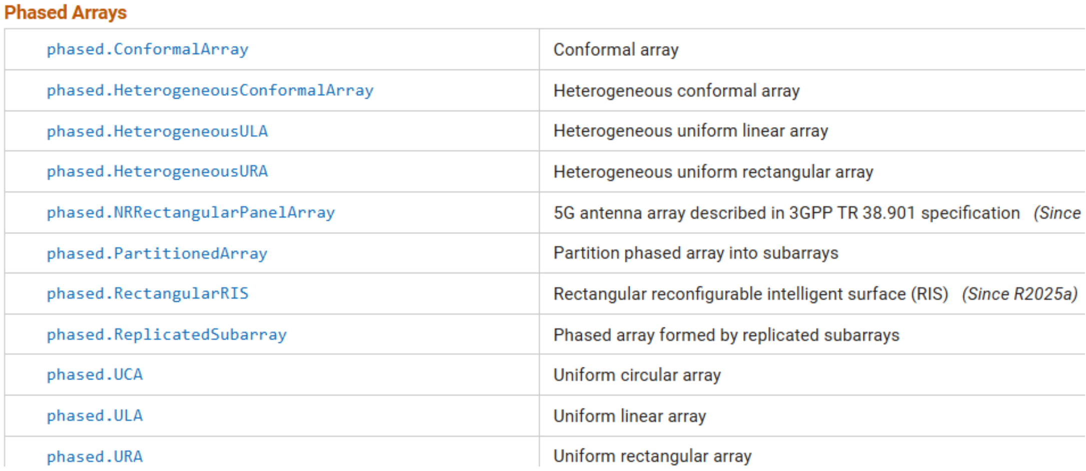
- Open the link below and view list
- Once **defined,** system objects can be **used** as functions





Antennas

phased.CardioidAntennaElement	Cardioid antenna element *(Since R2021b)*
phased.CosineAntennaElement	Cosine antenna element
phased.CrossedDipoleAntennaElement	Crossed-dipole antenna element
phased.CustomAntennaElement	Custom antenna element
phased.GaussianAntennaElement	Gaussian antenna element *(Since R2021b)*
phased.IsotropicAntennaElement	Isotropic antenna element
phased.NRAntennaElement	5G antenna element described in 3GPP TR 38.901 specificatic
phased.ShortDipoleAntennaElement	Short-dipole antenna element
phased.SincAntennaElement	Sinc antenna element *(Since R2021b)*

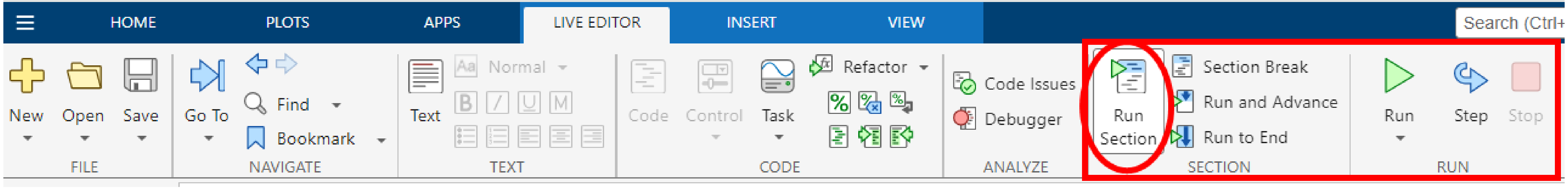


Phased Arrays

phased.ConformalArray	Conformal array
phased.HeterogeneousConformalArray	Heterogeneous conformal array
phased.HeterogeneousULA	Heterogeneous uniform linear array
phased.HeterogeneousURA	Heterogeneous uniform rectangular array
phased.NRRectangularPanelArray	5G antenna array described in 3GPP TR 38.901 specification *(Since*
phased.PartitionedArray	Partition phased array into subarrays
phased.RectangularRIS	Rectangular reconfigurable intelligent surface (RIS) *(Since R2025a)*
phased.ReplicatedSubarray	Phased array formed by replicated subarrays
phased.UCA	Uniform circular array
phased.ULA	Uniform linear array
phased.URA	Uniform rectangular array

mathworks.com/help/phased/referencelist.html
mathworks.com/help/matlab/matlab_prog/what-are-system-objects.html

MATLAB Live Scripts and Coding Practices



- Value-only arguments don’t need any modifiers
- Name-value pairs use property names

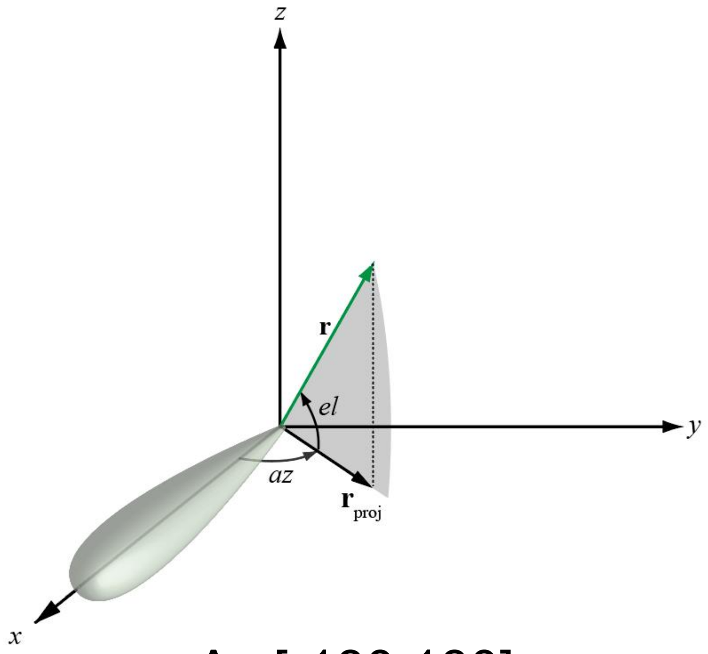
```
y = foo(arg1, ...                  % Value-only argument
            Name1=value1); % Name-value pair
```

- Name=value, and “Name”,value, and ‘Name’,value are all equivalent when specifying name-value pairs
  - Use only one style in a given function call

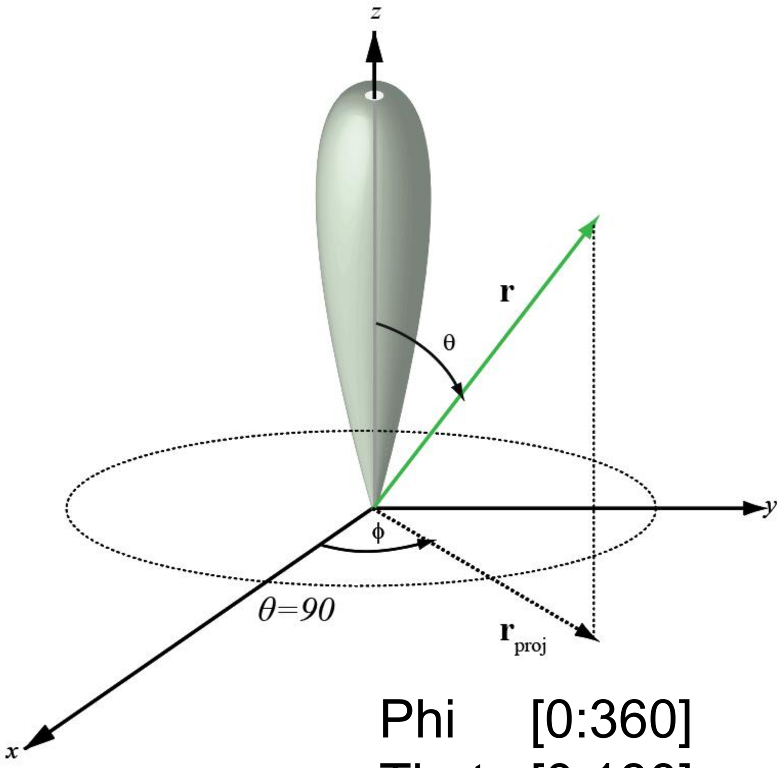
- Read the code comments! They contain helpful instructions to guide your code, including the usage of value-only arguments and name-value pairs.

Spherical Coordinate System in the Phased Array Toolbox

- Define a point in space with distance and two angles
- Phased Array Toolbox natively uses azimuth / elevation
- The toolbox provides **functions to convert between different spherical coordinate systems**



Az [-180:180]
El [-90:90]



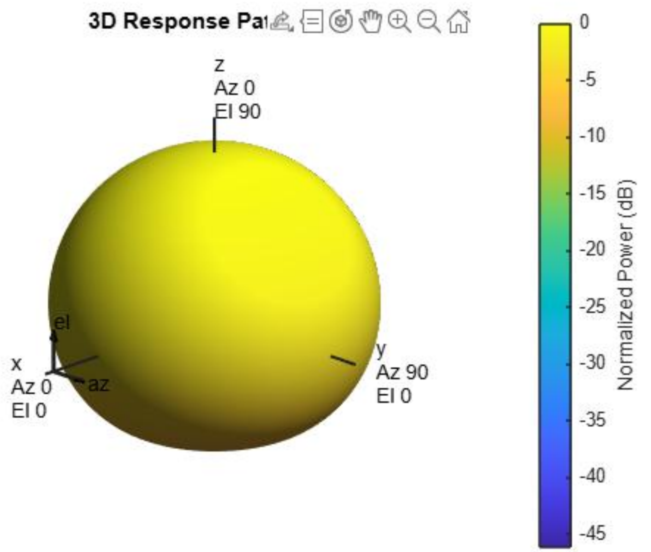
Phi   [0:360]
Theta [0:180]

$u = \sin\vartheta\cos\phi$
$v = \sin\vartheta\sin\phi$

mathworks.com/help/phased/ug/spherical-coordinates.html

Exercise 1: Create an array with custom antenna elements

- Steps:
  1. Create antenna element object
     1. Create a Cosine Antenna element object
     2. Import antenna element
     3. Plot element radiation pattern
  2. Construct 8x8 URA using these elements
  3. Visualize and analyze URA radiation pattern



- Related Resources

  mathworks.com/help/phased/ug/antenna-array-analysis-with-custom-radiation-pattern.html

WORK ON EXERCISE 1
(15 minutes)

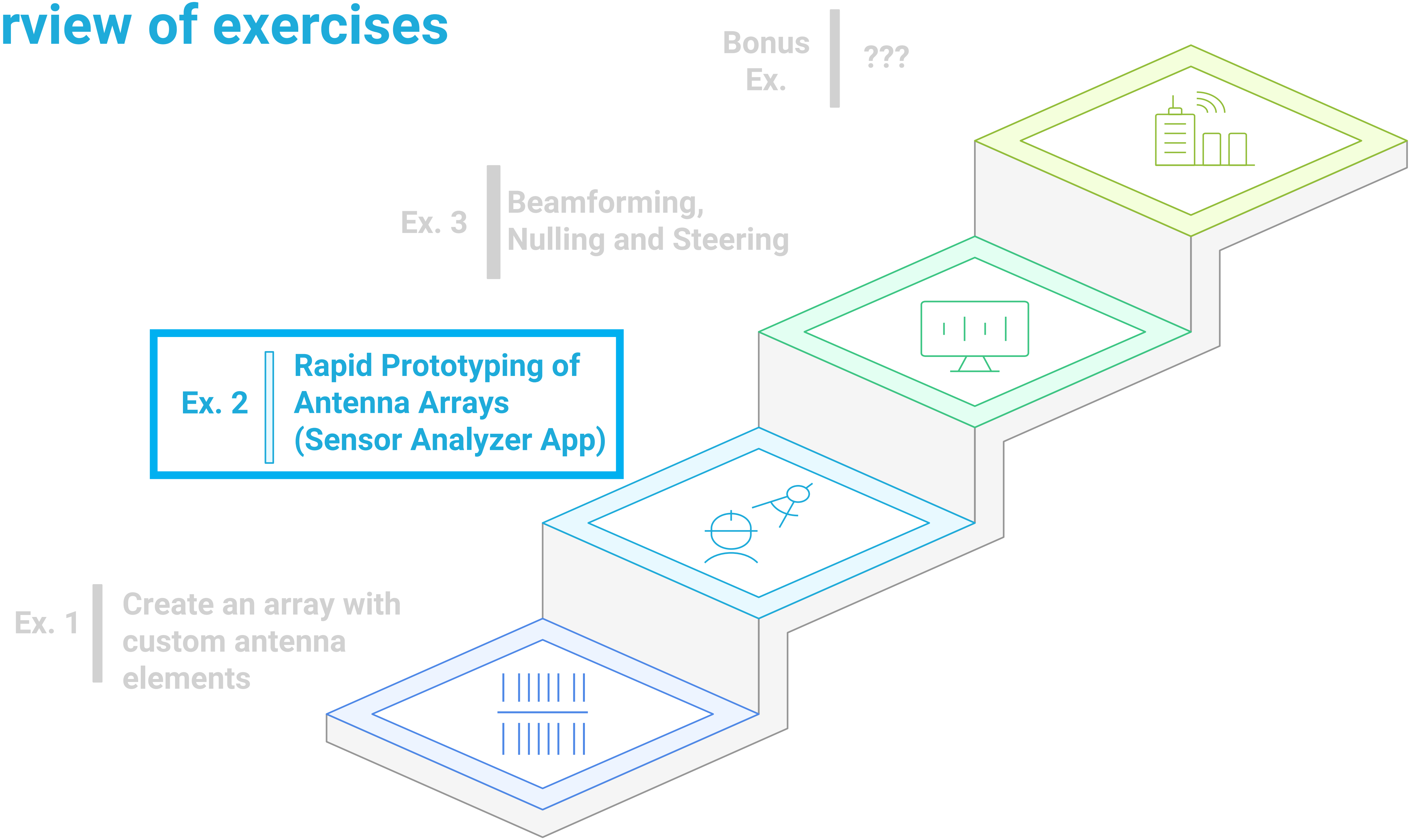
Let’s hear from you!

1. **Do you design antennas specifically for phased array systems?**
   Are you focused more on element-level design, array configuration, or both?

2. **What tools or software do you typically use for antenna or array design?**
   e.g., MATLAB, CST Studio, HFSS, in-house tools?

3. **How important is electromagnetic (EM) analysis in your workflow?**
   Do you perform full-wave simulations, or rely more on array-level approximations?

Overview of exercises



Bonus Ex. | ???

Ex. 3 | Beamforming, Nulling and Steering

Ex. 2 | Rapid Prototyping of Antenna Arrays (Sensor Analyzer App)

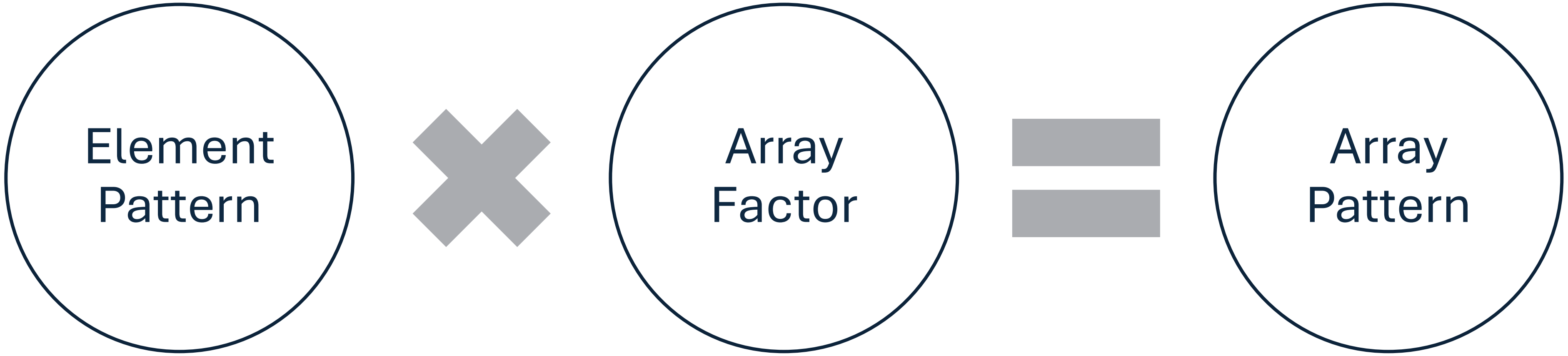
Ex. 1 | Create an array with custom antenna elements

2

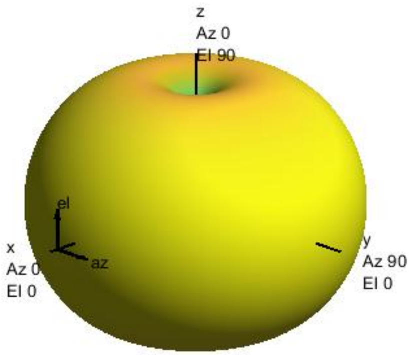
Rapid Prototyping of Antenna Arrays

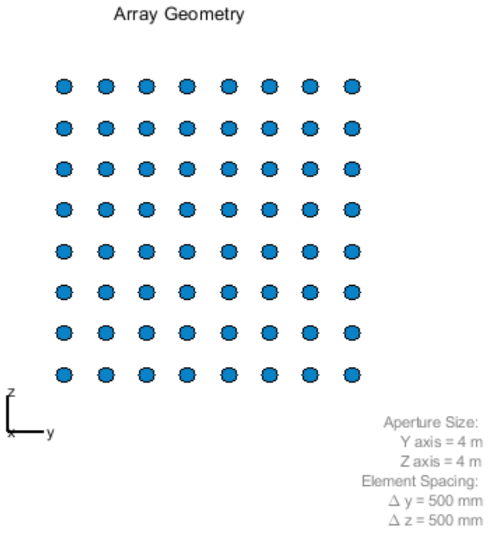
- Quick experiment with different array configurations
- View resultant beam pattern and 3D patterns
- Use the Sensor Array Analyzer App

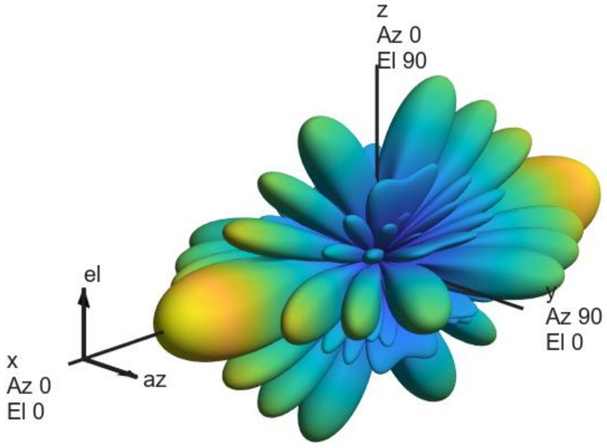
Array Patterns are Determined by Element Pattern and Spatial Configuration



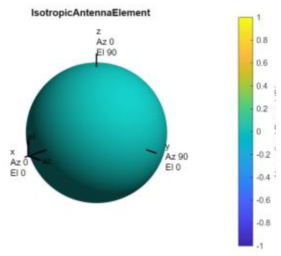
Element Pattern × Array Factor = Array Pattern



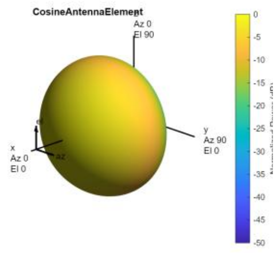




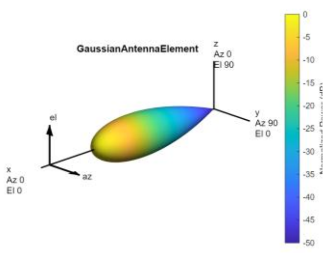
Options for element patterns



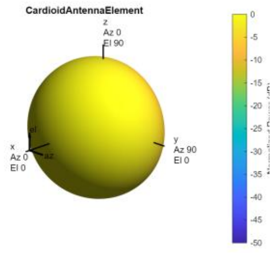
Isotropic Antenna



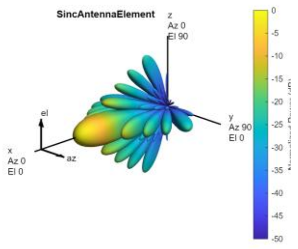
Cosine Antenna



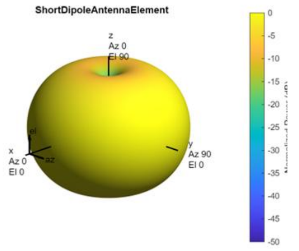
Gaussian Antenna



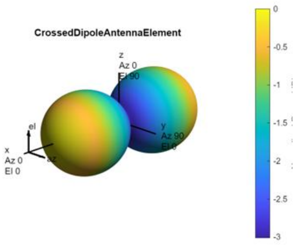
Cardioid Antenna



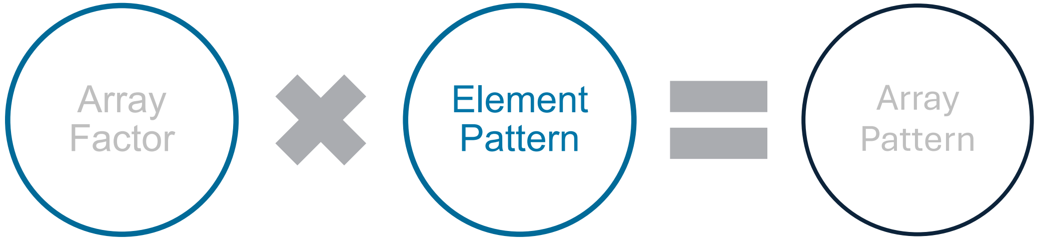
Sinc Antenna



Short Dipole

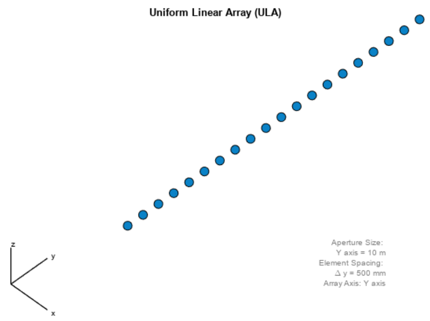


Crossed Dipole

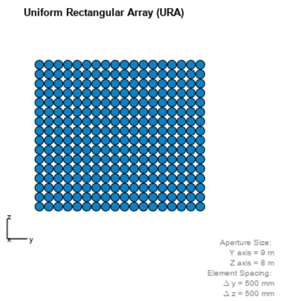


Array Factor × Element Pattern = Array Pattern

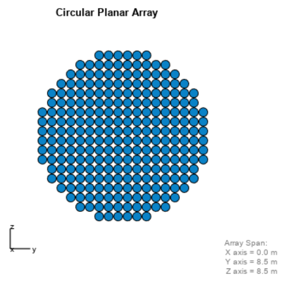
Options for Array Factor



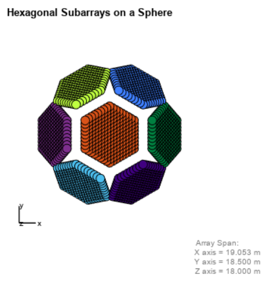
Uniform Linear Array



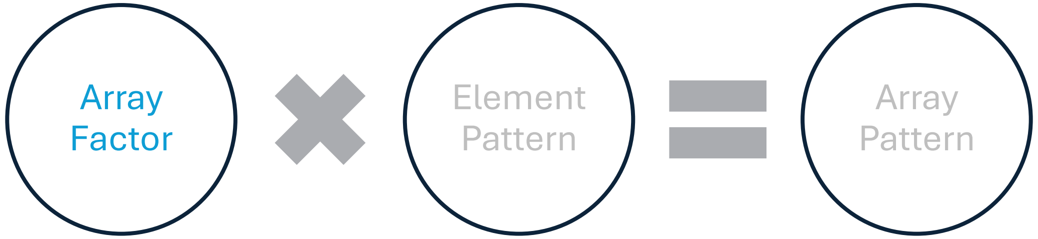
Uniform Rectangular Array



Circular Planar Array



Custom Array Geometry



Array Factor × Element Pattern = Array Pattern

Phased Array Antenna Performance Specifications

Parameter	
Operating Frequency Bands and Bandwidth	e.g. Ku-band (12-18 GHz) Ka-band (27-40GHz)
Antenna Architecture and Element Design	Antenna element, polarization, number of elements
Beamforming and Steering Performance	FOV, Beamwidth, Side Lobe Levels
Gain	Gain and directivity

Starlink User Terminal Phased Array Antenna

Specification	Starlink High Performance
Dimensions (L x W x H)	22" × 20" × 1.6" (575 x 511 x 41 mm) [18]
Weight	16 lbs (7.2 kg)
Average Power Consumption (Active)	110-150W
Idle Power Consumption	45W
Field of View	140°
Orientation	Fixed
Primary Frequency Band	Ku-band (12-18 GHz)
Estimated Element Count	~1000-1200

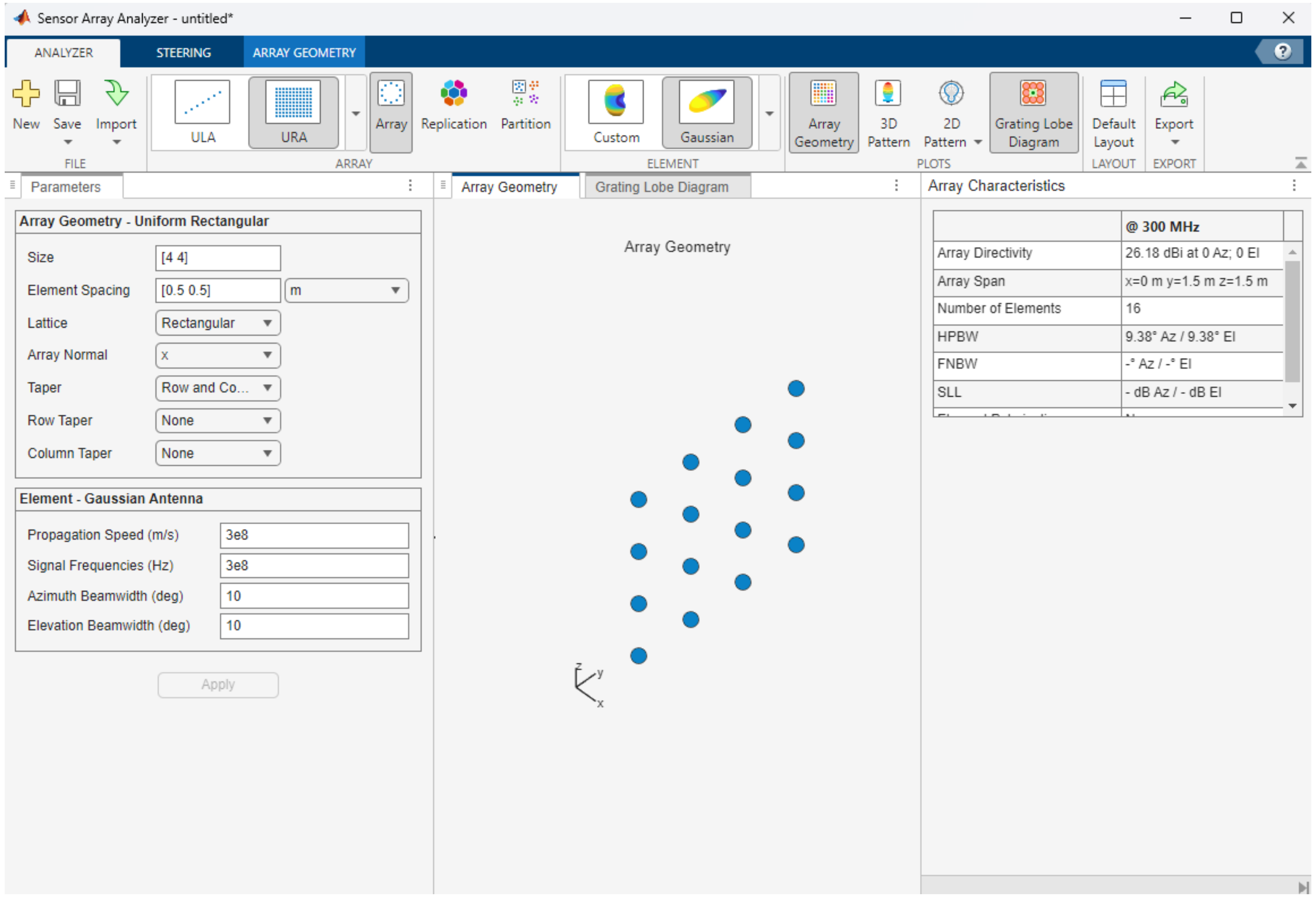
https://www.starlink.com/us/specifications?spec=2

Exercise 2: Rapid Prototyping of Antenna Arrays

Open the Sensor Array Analyzer App

- Apps tab from MATLAB
- Command Line:

```
>> sensorArrayAnalyzer
```



Sensor Array Analyzer - untitled*

Array Geometry - Uniform Rectangular

Size	[4 4]	
Element Spacing	[0.5 0.5]	m
Lattice	Rectangular	
Array Normal	x	
Taper	Row and Co...	
Row Taper	None	
Column Taper	None	

Element - Gaussian Antenna

Propagation Speed (m/s)	3e8
Signal Frequencies (Hz)	3e8
Azimuth Beamwidth (deg)	10
Elevation Beamwidth (deg)	10

Array Characteristics

	@ 300 MHz
Array Directivity	26.18 dBi at 0 Az; 0 El
Array Span	x=0 m y=1.5 m z=1.5 m
Number of Elements	16
HPBW	9.38° Az / 9.38° El
FNBW	-° Az / -° El
SLL	- dB Az / - dB El

Exercise 2: Rapid Prototyping of Antenna Arrays

Steps:

1. Experiment with different array factors and element patterns.
   - View 3D and 2D patterns

2. Design and export phased array with the following configuration:
   - *Antenna Elements:*
     - *Isotropic*
     - *Back Baffled*
   - *URA 8 x 8*
   - *Spacing ½ λ*
   - *Signal Freq 2.5 GHz*
   - *Azimuth Steering Angle 15*

3. Export to MATLAB

4. Extra Credit:
   - *Find the effect of grating lobes and tapering*
   - *Partition array into subarrays*

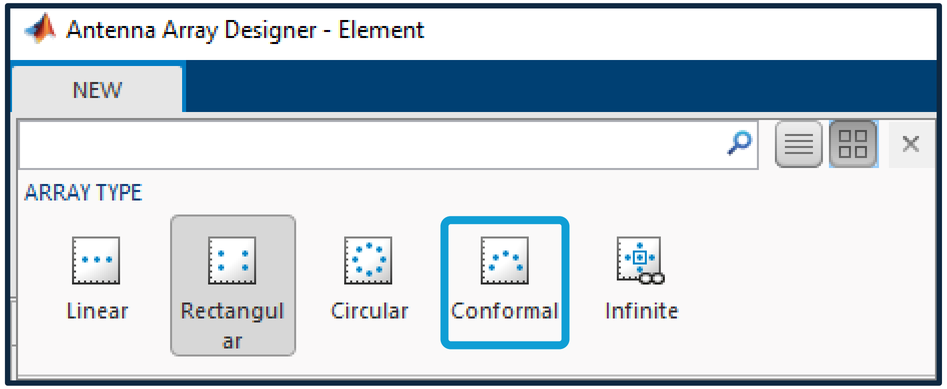
WORK ON EXERCISE 2 (15 minutes)

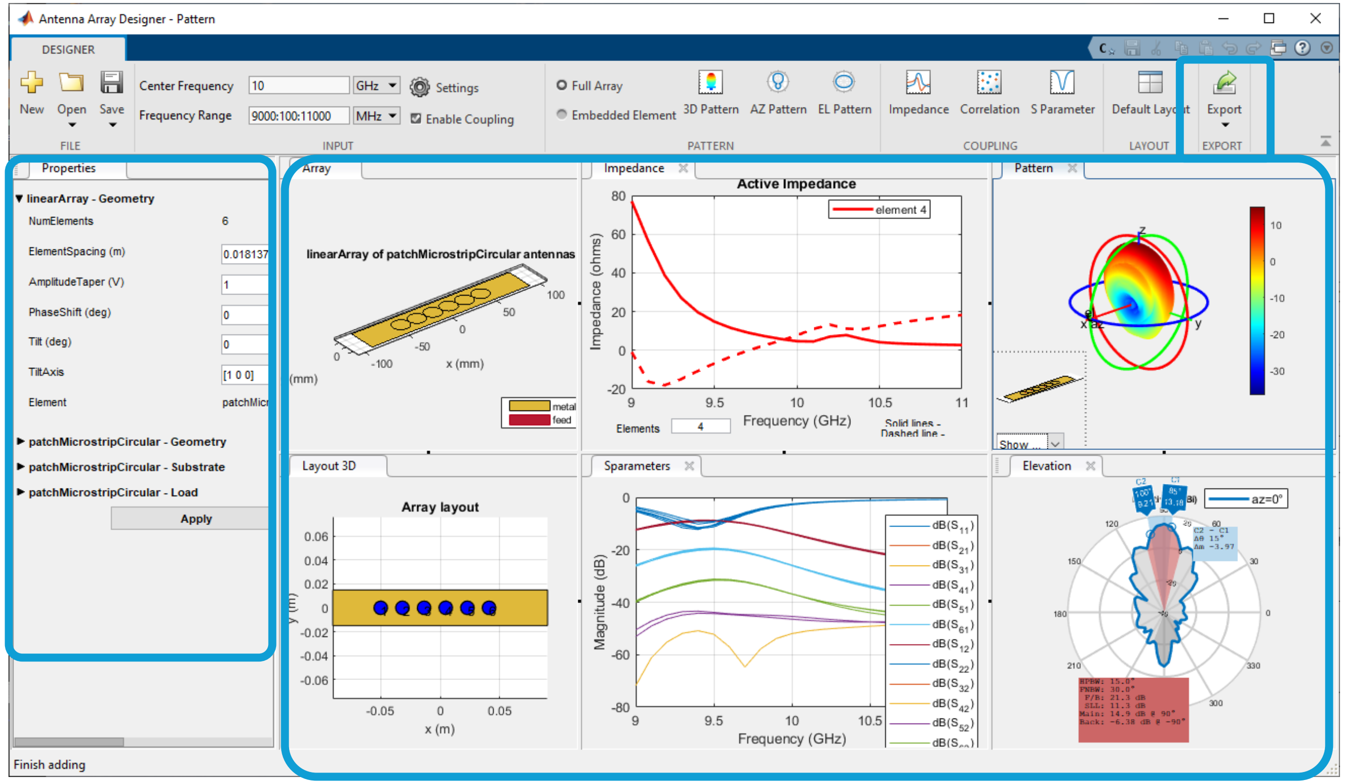
Related Resources

Antenna Array Designer

Antenna Toolbox:

- Full-wave and hybrid electromagnetic solvers for antenna elements and arrays
- Do port, surface, and field analysis, antenna placement studies, and radar cross section (RCS) calculations.
- Export designed antennas for fabrication using CAD and Gerber files





Antenna Array Designer - Pattern

Center Frequency: 10 GHz

Frequency Range: 9000:100:11000 MHz

☑ Enable Coupling

○ Full Array
○ Embedded Element

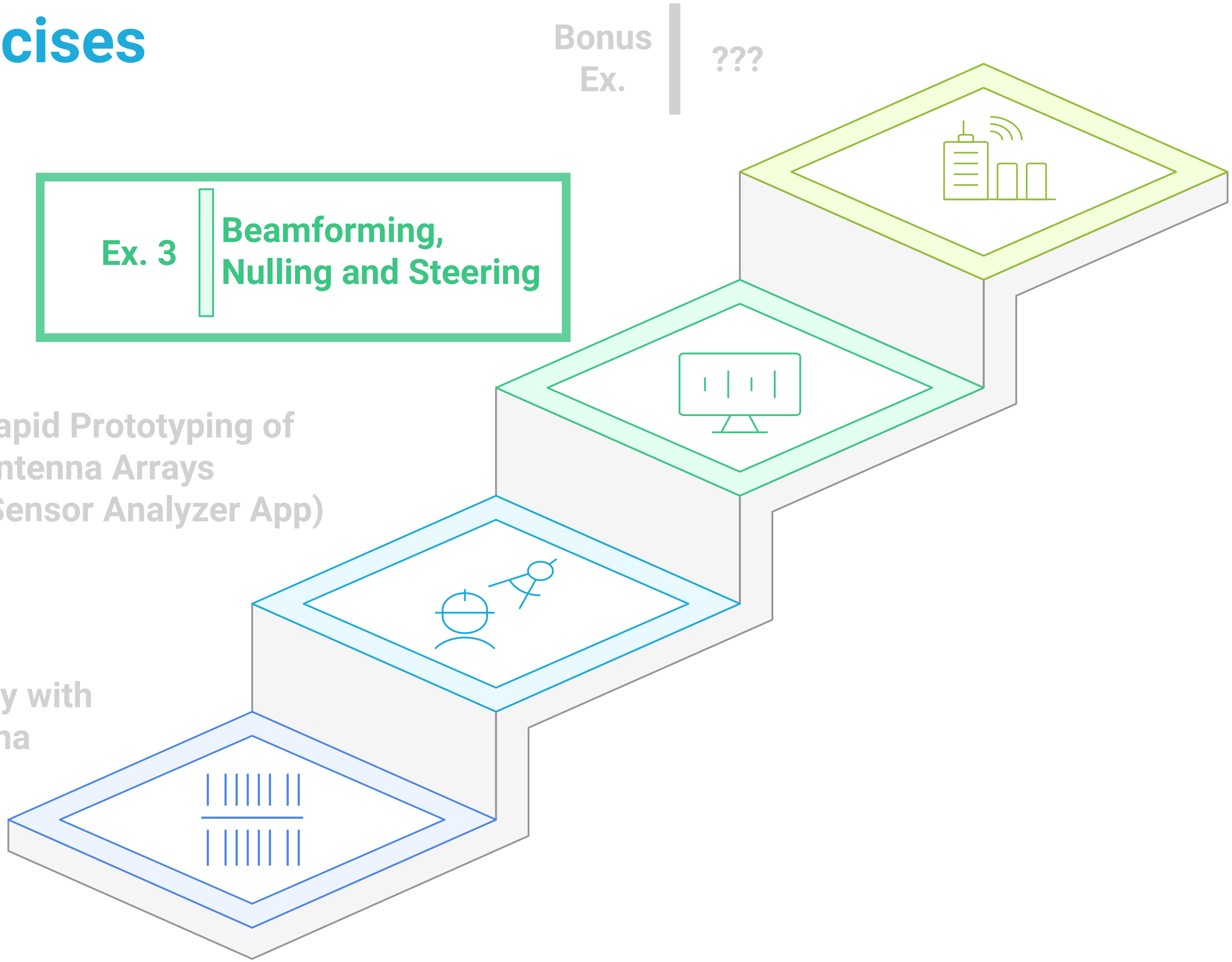
linearArray - Geometry

NumElements	6
ElementSpacing (m)	0.018137
AmplitudeTaper (V)	1
PhaseShift (deg)	0
Tilt (deg)	0
TiltAxis	[1 0 0]
Element	patchMicr

▶ patchMicrostripCircular - Geometry
▶ patchMicrostripCircular - Substrate
▶ patchMicrostripCircular - Load

Finish adding

Overview of exercises



Bonus Ex. | ???

Ex. 3 | Beamforming, Nulling and Steering

Ex. 2 | Rapid Prototyping of Antenna Arrays (Sensor Analyzer App)

Ex. 1 | Create an array with custom antenna elements

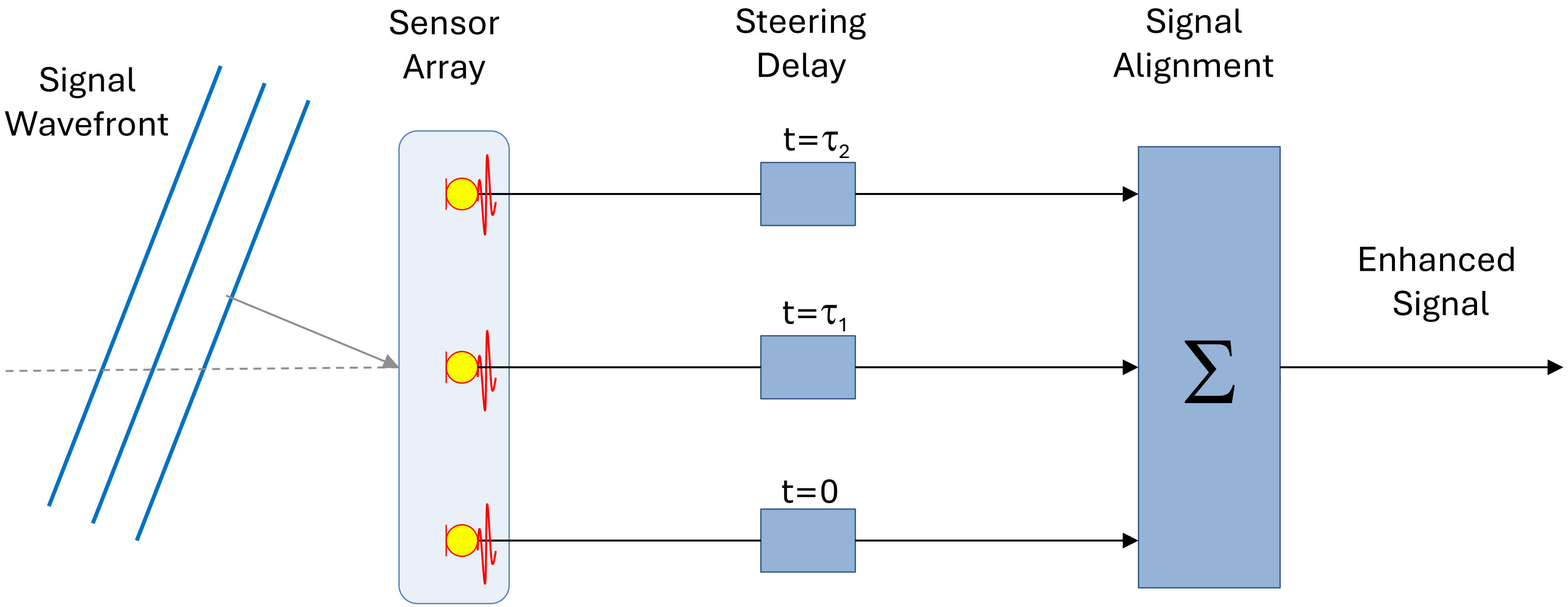
3

Beamforming, Nulling and Steering

- Calculate weights for a ULA
- Meet given requirements for array performance
- Get familiar with minimum variance optimization
- Perform beamforming, nulling and steering

What is Beamforming and Nulling?

- Adjust weights of array elements to enhance signals from:
  - desired directions (steering)
  - suppress interference/noise from others (nulling)
- Conventional (fixed weights) vs Adaptive (respond to environment)
- Narrowband (constant phase shifts) vs Wideband (time delays)



www.mathworks.com/help/phased/beamforming.html

Supported beamformers in MATLAB

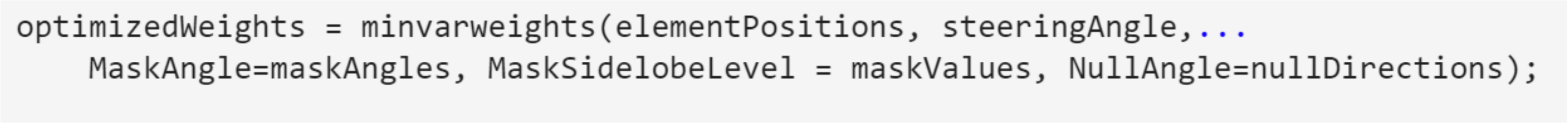
	Beamformer Name	Bandwidth	Processing Domain
Conventional	phased.PhaseShiftBeamformer	Narrowband	Time domain
	phased.TimeDelayBeamformer	Wideband	Time domain
	phased.SubbandPhaseShiftBeamformer	Wideband	Frequency domain
Adaptive	phased.LCMVBeamformer	Narrowband	Frequency domain
	phased.MVDRBeamformer	Narrowband	Frequency domain
	phased.FrostBeamformer	Wideband	Time domain
	phased.GSCBeamformer	Wideband	Time domain
	phased.TimeDelayLCMVBeamformer	Wideband	Time domain
	phased.SubbandMVDRBeamformer	Wideband	Frequency domain

mathworks.com/help/phased/ug/beamforming-concepts.html

Minimum Variance Beamforming: Adaptive Array Weight Design

- Computes array weights that minimize output noise/interference while preserving gain in desired direction
- Mathematically it is an optimization problem
- Narrower main lobe, deeper nulls compared to conventional beamforming
- Adaptive: responds to interference environment
- When sidelobes are suppressed, the main lobe widens

In MATLAB:

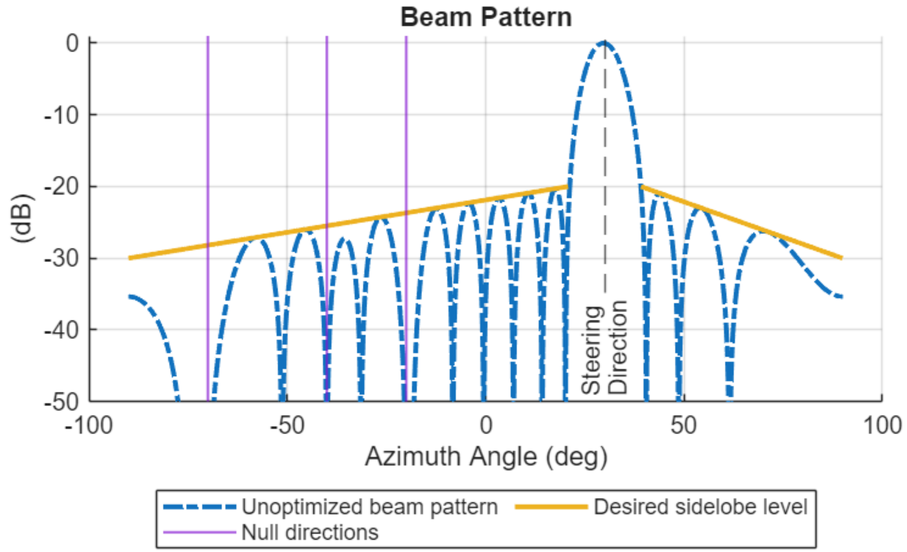


```
optimizedWeights = minvarweights(elementPositions, steeringAngle,...
    MaskAngle=maskAngles, MaskSidelobeLevel = maskValues, NullAngle=nullDirections);
```

Exercise 3: Beamforming, Nulling and Steering

- Steps:
  - Use minimum-variance algorithm to calculate array weights to meet these requirements:

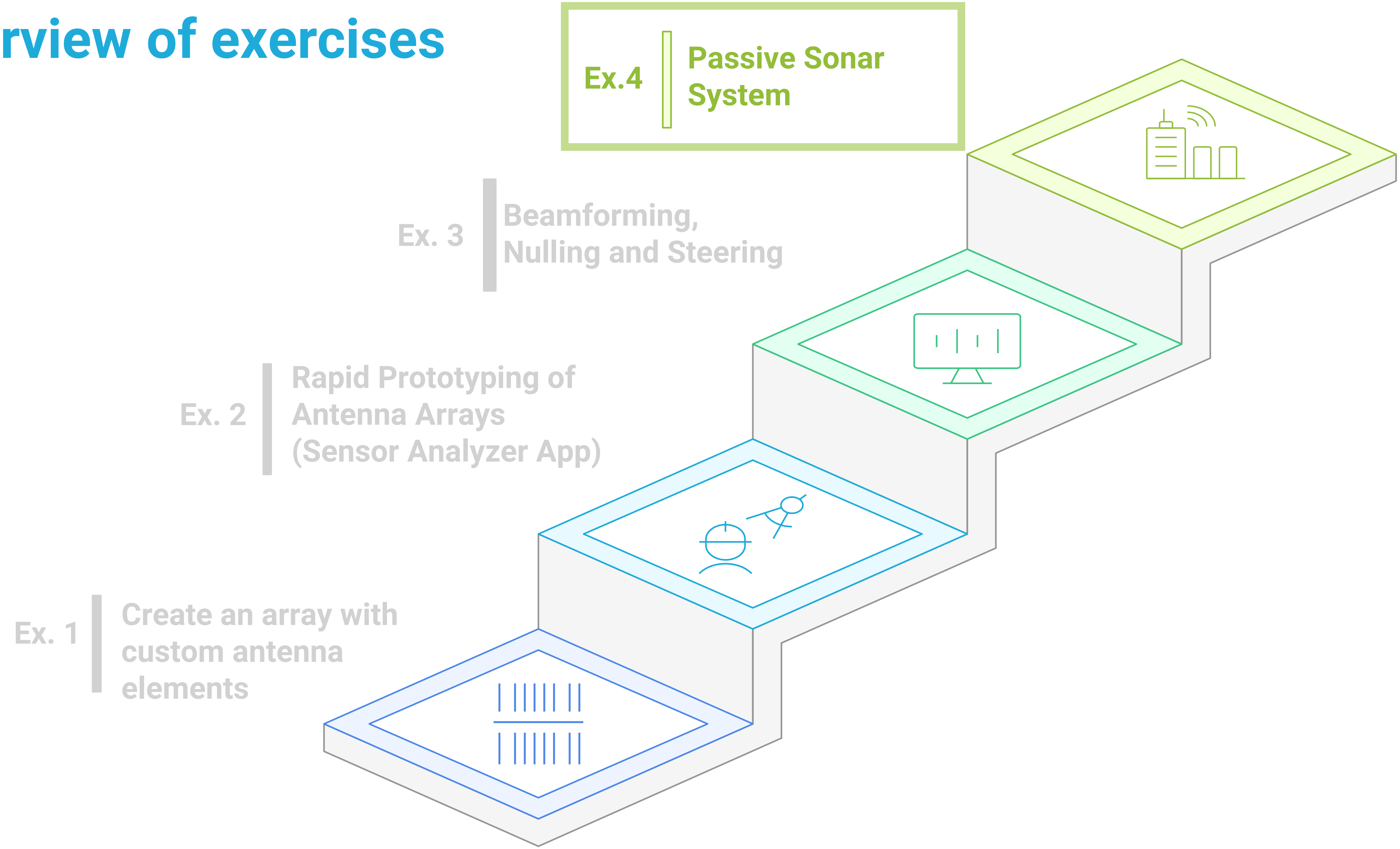
Specifications	Details
Configuration and Size	Uniform Linear Array with 16 elements
Operating frequency	3 GHz
Main beam angle / Broad side (Az)	30°
Null angles	-70°, -40°, -20°
Side Lobe Level (SLL) Suppression	Linear from -20dB to -30dB (symmetric)



  - Plot and view the resultant beam pattern

WORK ON EXERCISE 3 (15 minutes)

Overview of exercises



Ex.4 Passive Sonar System

Ex. 3 Beamforming, Nulling and Steering

Ex. 2 Rapid Prototyping of Antenna Arrays (Sensor Analyzer App)

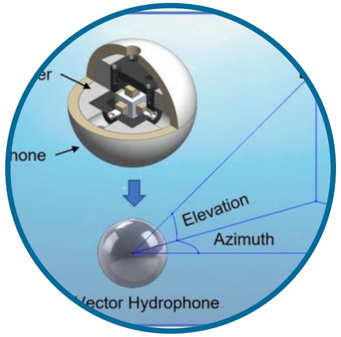
Ex. 1 Create an array with custom antenna elements

4

Modeling Sonar Systems

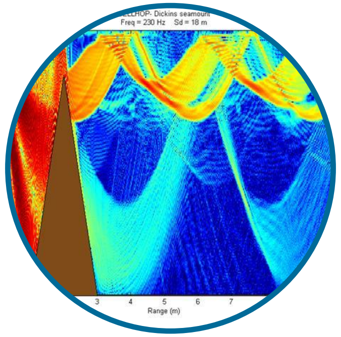
- Calculate weights for a ULA
- Meet given requirements for array performance
- Get familiar with minimum variance optimization
- Perform beamforming, nulling and steering

Application Areas - Underwater Acoustics and Marine Systems



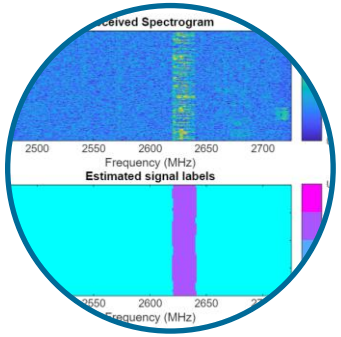
Underwater Acoustic Elements and Ranging

- Sonar array modeling
- Beamforming and DOA
- Waveform design, match filtering
- Active and passive sonar modeling



Underwater Acoustic Propagation Models

- Multipath, Attenuation, spreading
- Propagation Models – Bellhop
- Time-varying / statistical models

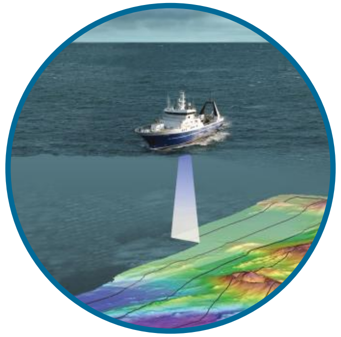


Signal Processing and Comms

- Acoustic modem modeling (modulator, demodulator)
- Error correction
- Signal detection and classification

Marine Vehicles, Robotics and Navigation

- Vehicle dynamics (AUVs, UUVs, surface vessels)
- Navigation & sensor fusion
- Path planning, SLAM, and guidance/control systems
- Hydrodynamic effects (drag, added mass, thrusters)



Ocean Environment and Mapping

- Ocean surface modeling
- Scenario Modeling
- Bathymetry and mapping
- Wave models, ocean currents

Existing Core Sonar Support

Workflow	Available functions/objects

Define transducer arrays (ULA/UCA) with acoustic elements	phased.ULA, phased.UCA, phased.ConformalArray;
	phased.IsotropicHydrophone, phased.IsotropicProjector;
	pattern, plotResponse

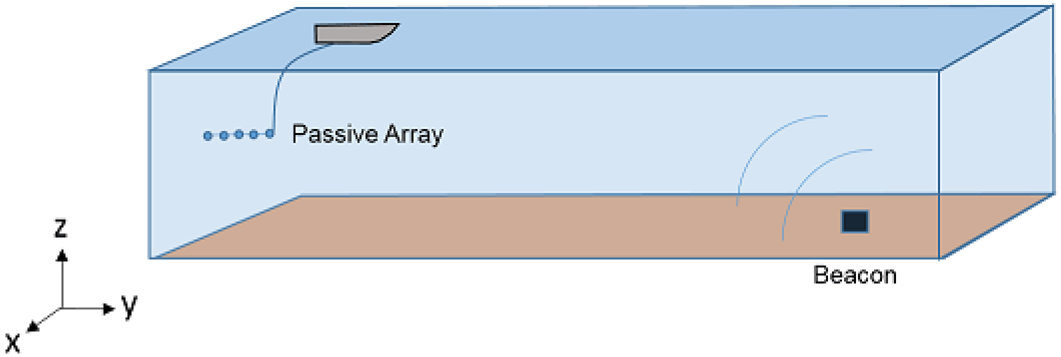
Design & simulate waveforms (LFM/NLFM/FMCW, custom)	phased.LinearFMWaveform, phased.NonlinearFMWaveform, phased.FMCWWaveform; dsp.MatchedFilter; Signal Processing tools for custom pulses
Underwater Channels:	IsoSpeedUnderwaterpaths, phased.MultipathChannel, phased.UnderwaterRadiatedNoise

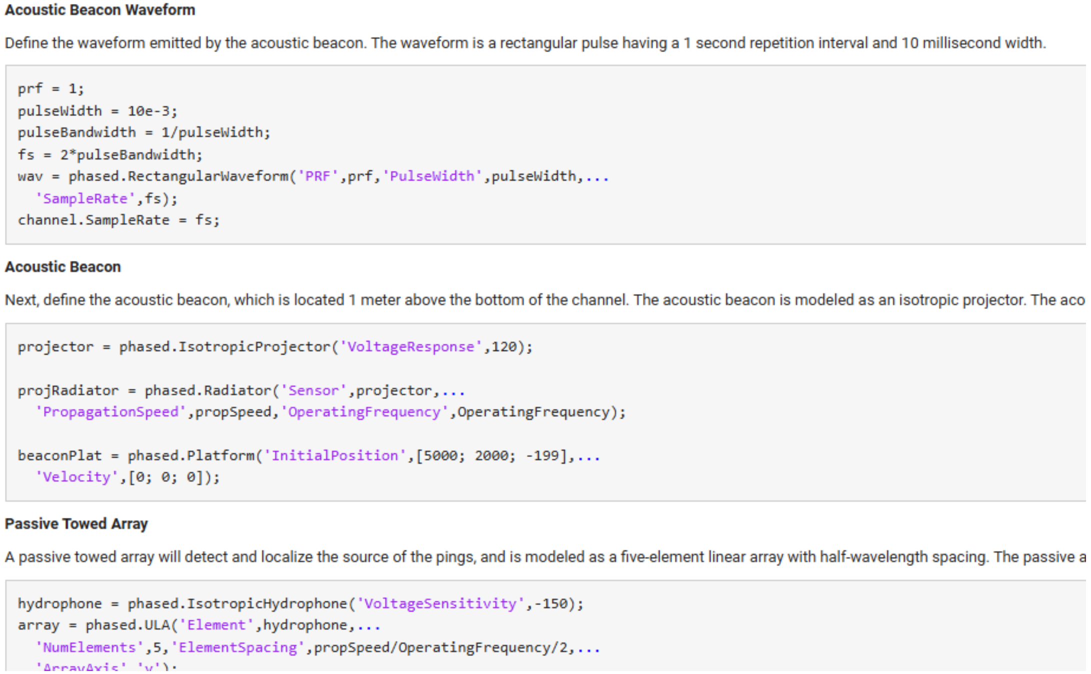
Scenario setup: targets/clutter, bathymetry/context; compute performance with sonar equations	phased.Backscatterer, phased.Platform;
	Mapping: readgeoraster, geoshow, geoplot;
	Performance: sonarEquation, sonareqsnr, range2tl

Process returns: beamforming (LCMV/MVDR), DOA (MUSIC/Root-MUSIC), tracking	Beamforming: phased.LCMVBeamformer, phased.TimeDelayBeamformer, phased.PhaseShiftBeamformer;
	DOA: phased.MUSICEstimator, phased.RootMUSICEstimator;
	Tracking: trackerGNN, trackerTOMHT, trackerJPDA, trackingKF, trackingEKF

www.mathworks.com/help/phased/beamforming.html

Example: Locating an Acoustic Beacon with a Passive Sonar System





Define the Acoustic Beacon and Passive Array

Acoustic Beacon Waveform

Define the waveform emitted by the acoustic beacon. The waveform is a rectangular pulse having a 1 second repetition interval and 10 millisecond width.

```
prf = 1;
pulseWidth = 10e-3;
pulseBandwidth = 1/pulseWidth;
fs = 2*pulseBandwidth;
wav = phased.RectangularWaveform('PRF',prf,'PulseWidth',pulseWidth,...
  'SampleRate',fs);
channel.SampleRate = fs;
```

Acoustic Beacon

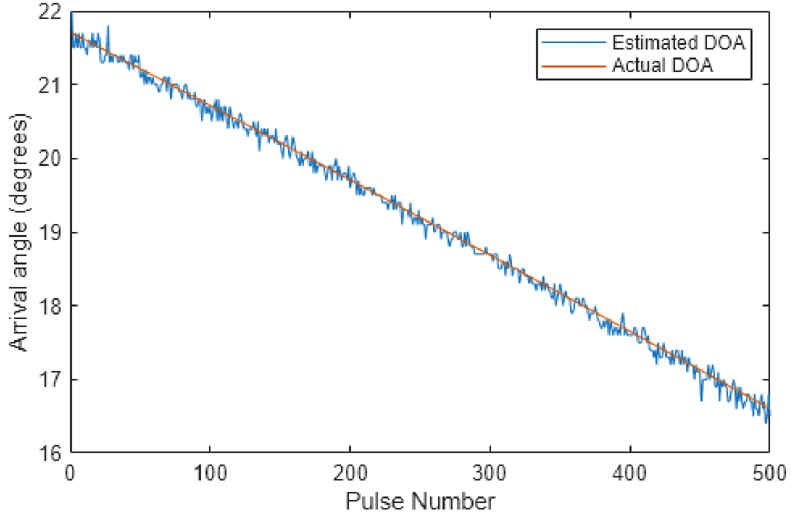
Next, define the acoustic beacon, which is located 1 meter above the bottom of the channel. The acoustic beacon is modeled as an isotropic projector. The aco

```
projector = phased.IsotropicProjector('VoltageResponse',120);

projRadiator = phased.Radiator('Sensor',projector,...
  'PropagationSpeed',propSpeed,'OperatingFrequency',OperatingFrequency);

beaconPlat = phased.Platform('InitialPosition',[5000; 2000; -199],...
  'Velocity',[0; 0; 0]);
```

Passive Towed Array

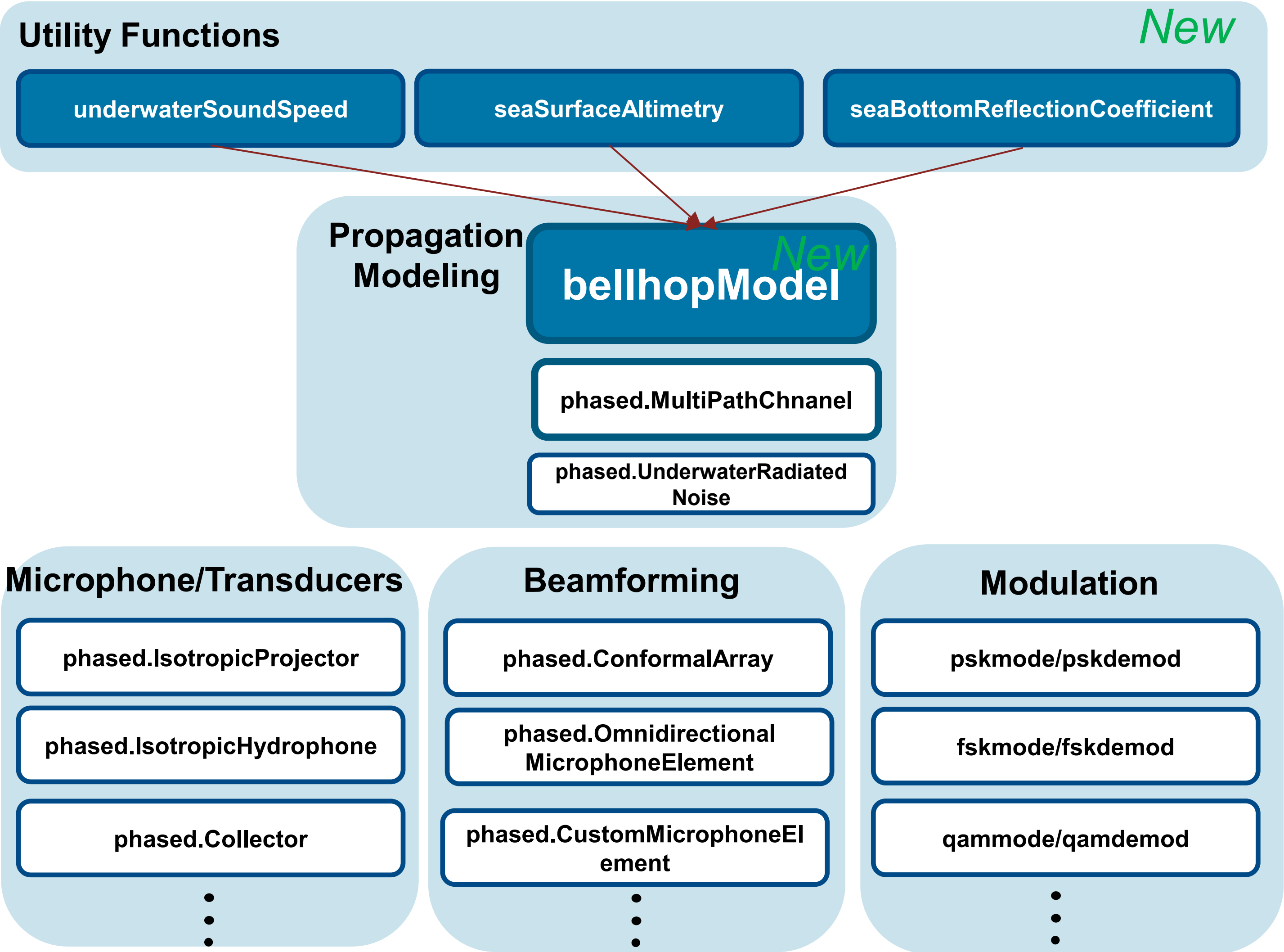
A passive towed array will detect and localize the source of the pings, and is modeled as a five-element linear array with half-wavelength spacing. The passive a

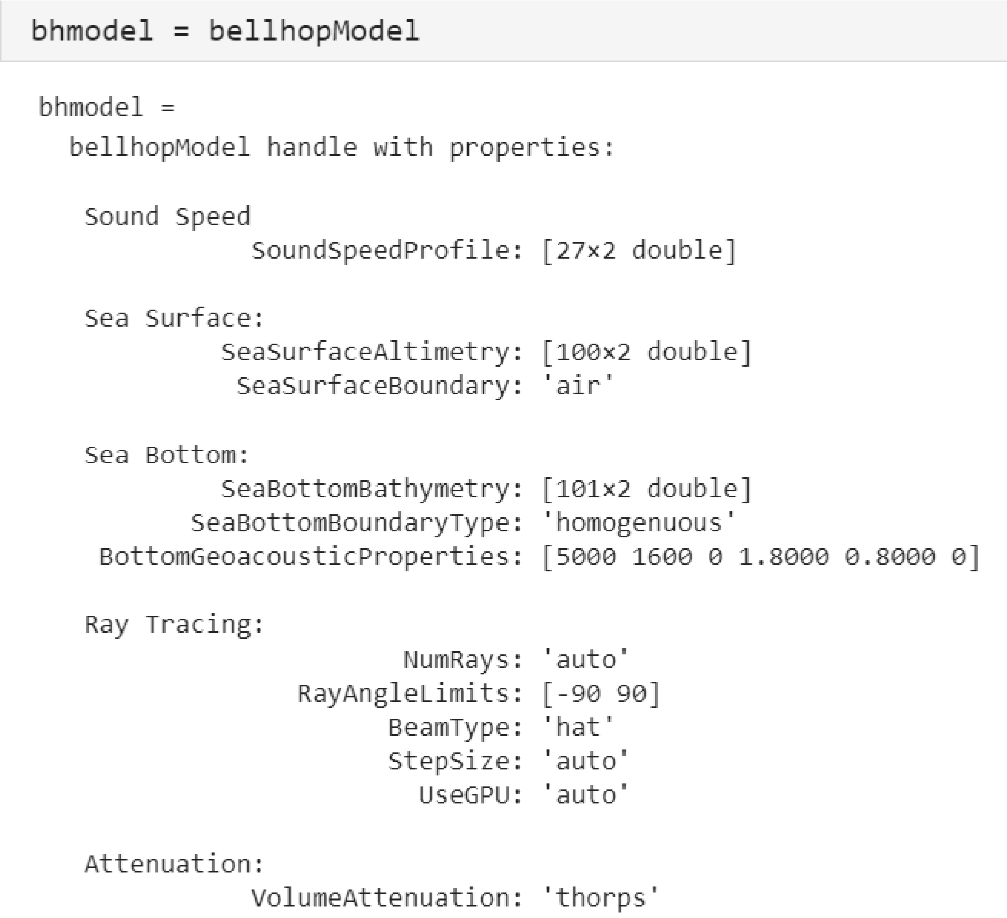
```
hydrophone = phased.IsotropicHydrophone('VoltageSensitivity',-150);
array = phased.ULA('Element',hydrophone,...
  'NumElements',5,'ElementSpacing',propSpeed/OperatingFrequency/2,...
  'ArrayAxis','y');
```



mathworks.com/help/phased/ug/locating-an-acoustic-beacon-with-a-passive-sonar-system.html

A New Bellhop interface and utility functions to integrate Bellhop in MATLAB ecosystem

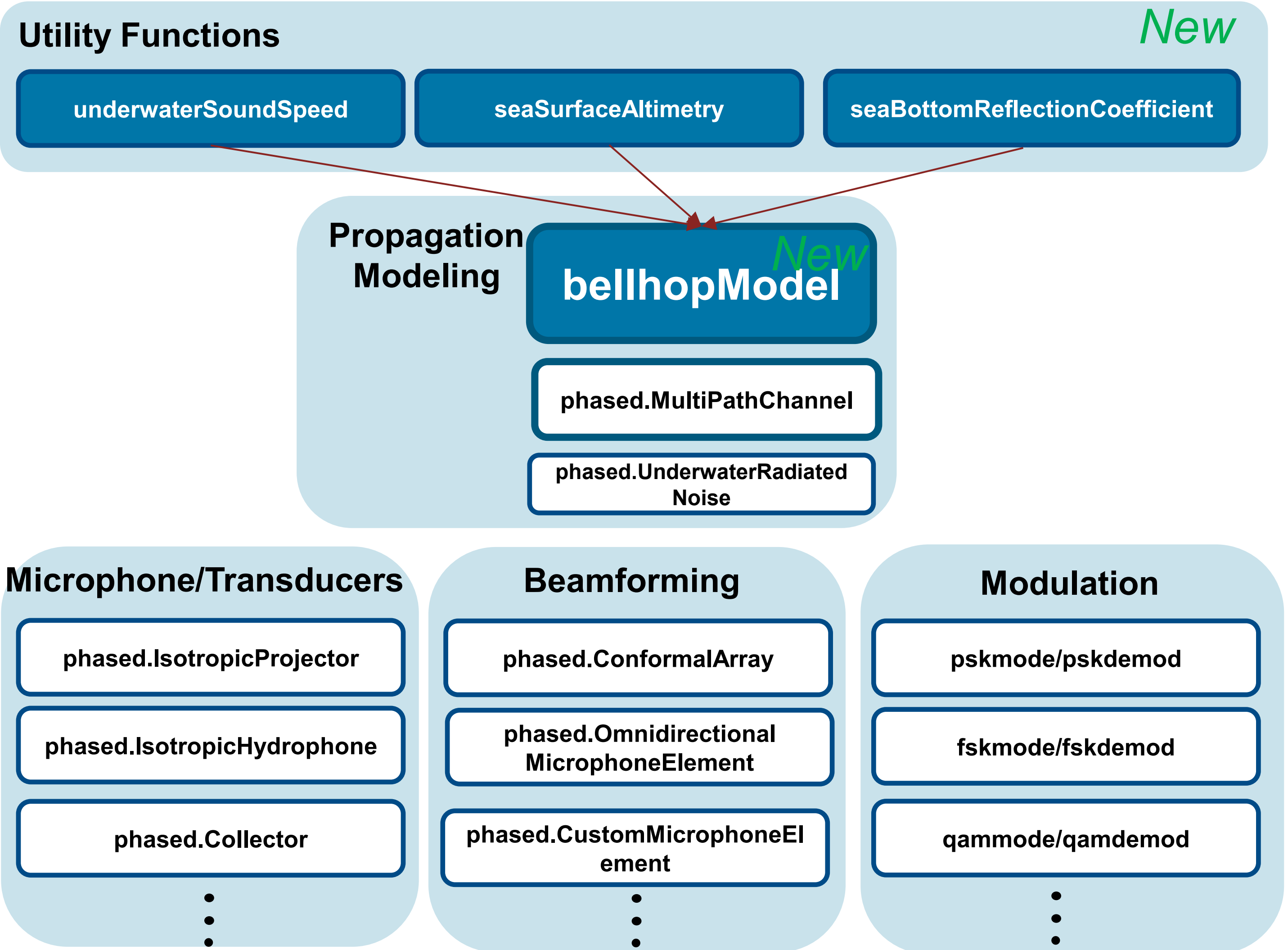




```
bhmodel = bellhopModel
```

```
bhmodel = 
  bellhopModel handle with properties:

   Sound Speed
              SoundSpeedProfile: [27×2 double]

   Sea Surface:
            SeaSurfaceAltimetry: [100×2 double]
             SeaSurfaceBoundary: 'air'

   Sea Bottom:
            SeaBottomBathymetry: [101×2 double]
          SeaBottomBoundaryType: 'homogenuous'
    BottomGeoacousticProperties: [5000 1600 0 1.8000 0.8000 0]

   Ray Tracing:
                        NumRays: 'auto'
                 RayAngleLimits: [-90 90]
                       BeamType: 'hat'
                       StepSize: 'auto'
                         UseGPU: 'auto'

   Attenuation:
              VolumeAttenuation: 'thorps'
```

A New Bellhop interface and utility functions to integrate Bellhop in MATLAB ecosystem



Utility Functions

New

underwaterSoundSpeed

seaSurfaceAltimetry

seaBottomReflectionCoefficient

Propagation Modeling

bellhopModel

New

phased.MultiPathChannel

phased.UnderwaterRadiated Noise

Microphone/Transducers

phased.IsotropicProjector

phased.IsotropicHydrophone

phased.Collector

⋮

Beamforming

phased.ConformalArray

phased.Omnidirectional MicrophoneElement

phased.CustomMicrophoneEl ement

⋮

Modulation

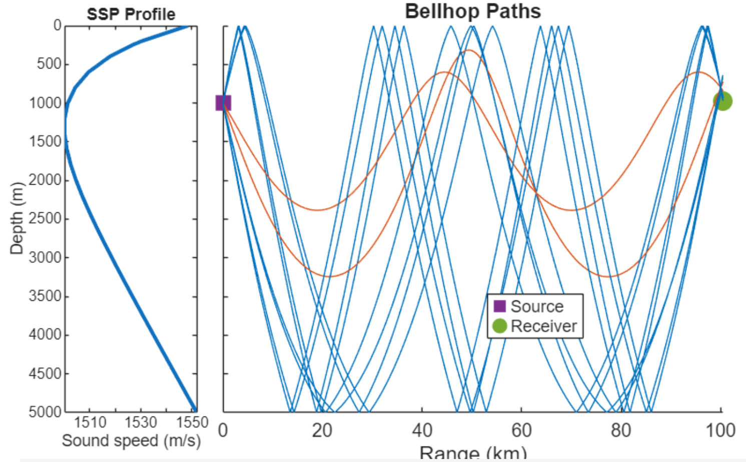
pskmode/pskdemod

fskmode/fskdemod

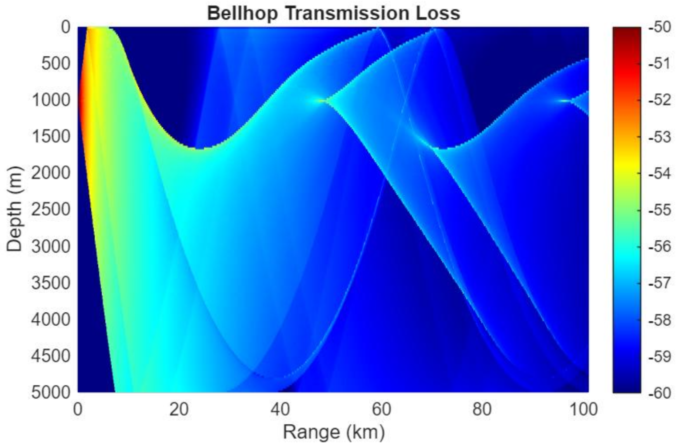
qammode/qamdemod

⋮

```
plotEigenRays(bhmodel,srcLocation,rxLocation,fc)
```

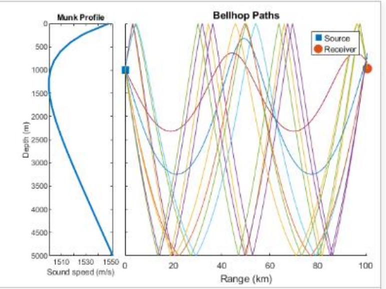


```
plotTransmissionLoss(bhmodel,srcLocation,rxLocation,fc)
```



Proposal: R2026b Examples

Active Sonar with Inline Bellhop Example



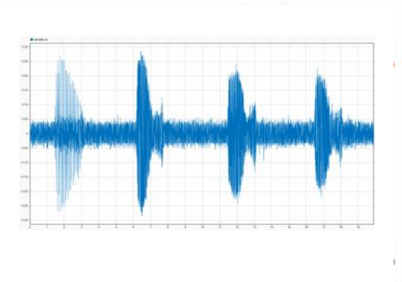
Underwater Target Detection with an Active Sonar System

Simulate an active monostatic sonar scenario with two targets. The sonar system consists of an isotropic projector array and a single...

Products

- UWA Channel Support Package
- Phased Array System Toolbox

DL-based underwater acoustic signal classification with Bellhop Example



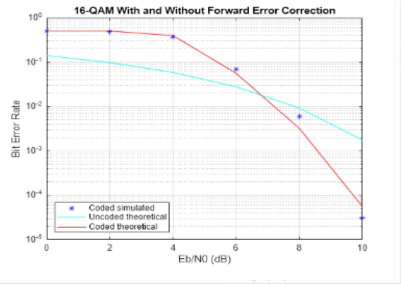
Underwater Acoustic Signal Classification

Preprocess underwater acoustic signals and pass them through Bellhop channel model. Use deep neural network to classify …

Products

- UWA Channel Support Package
- Phased Array System Toolbox
- Audio/Signal Toolbox?
- Deep Learning Toolbox

Underwater Communications Link BER Simulation Example



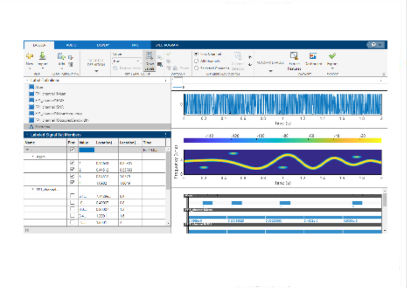
Underwater Communications Link Simulation

Use Bellhop channel model to simulate an underwater communications link with error correction codes.

Products

- UWA Channel Support Package
- Communications Toolbox

Label Underwater Acoustic Signals with Signal Labeler



Label Underwater Acoustic Signals with Signal Labeler

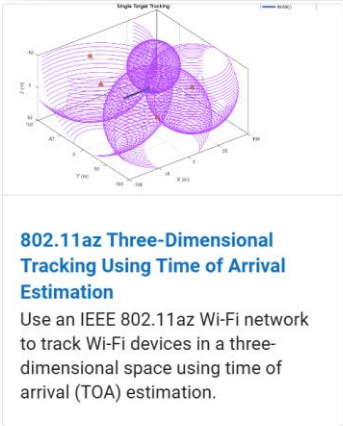
Use signal labeler app to automatically label underwater acoustic spectrograms.

Products

- Signal Processing Toolbox

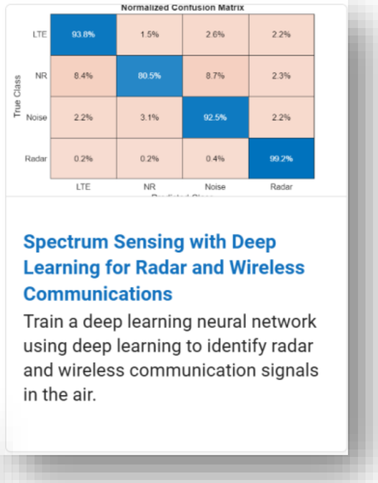
Other Key Topics (Not Covered Today)

- Direction of Arrival Estimation and STAP
- Detection, Range and Doppler Estimation
- Waveform Design and Signal Synthesis
- Radar and Wireless Coexistence
- Localization



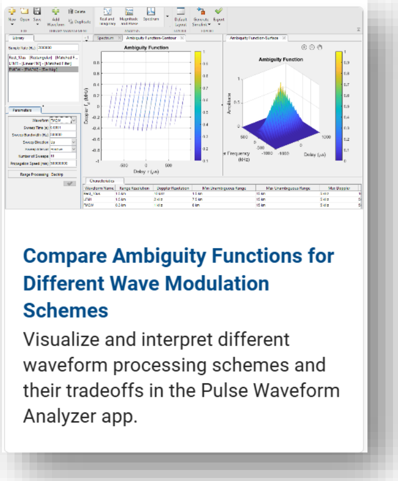
802.11az Three-Dimensional Tracking Using Time of Arrival Estimation

Use an IEEE 802.11az Wi-Fi network to track Wi-Fi devices in a three-dimensional space using time of arrival (TOA) estimation.



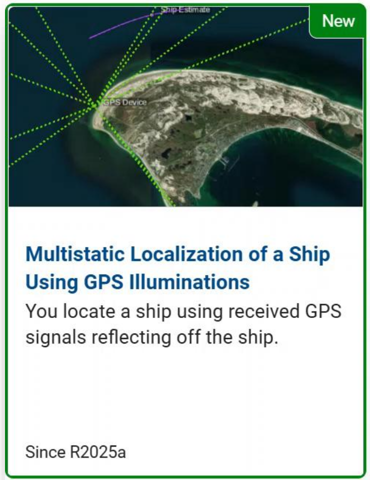
Spectrum Sensing with Deep Learning for Radar and Wireless Communications

Train a deep learning neural network using deep learning to identify radar and wireless communication signals in the air.



Compare Ambiguity Functions for Different Wave Modulation Schemes

Visualize and interpret different waveform processing schemes and their tradeoffs in the Pulse Waveform Analyzer app.



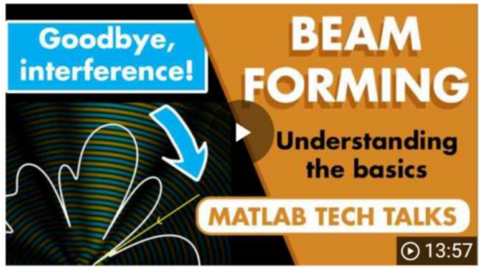
Multistatic Localization of a Ship Using GPS Illuminations

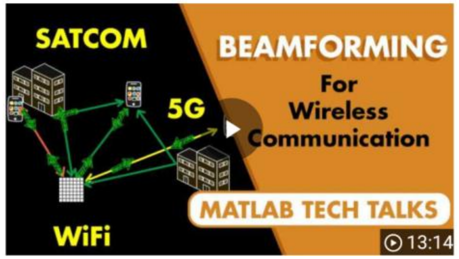
You locate a ship using received GPS signals reflecting off the ship.

Since R2025a

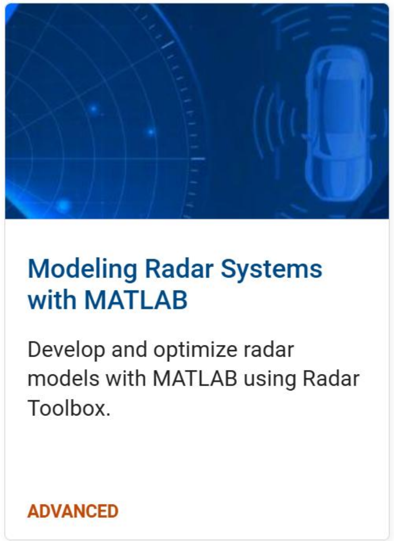
Related Resources:

- Get Started with the Phased Array System Toolbox
- MATLAB Tech Talks Series on Phased Arrays and Beamforming





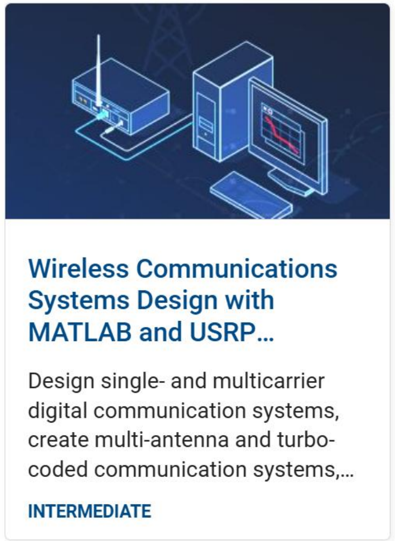
- 2-Day Course: Modeling Radar Systems with MATLAB



Modeling Radar Systems with MATLAB

Develop and optimize radar models with MATLAB using Radar Toolbox.

ADVANCED



Wireless Communications Systems Design with MATLAB and USRP...

Design single- and multicarrier digital communication systems, create multi-antenna and turbo-coded communication systems,...

INTERMEDIATE

Summary

Skills You Gained Today:

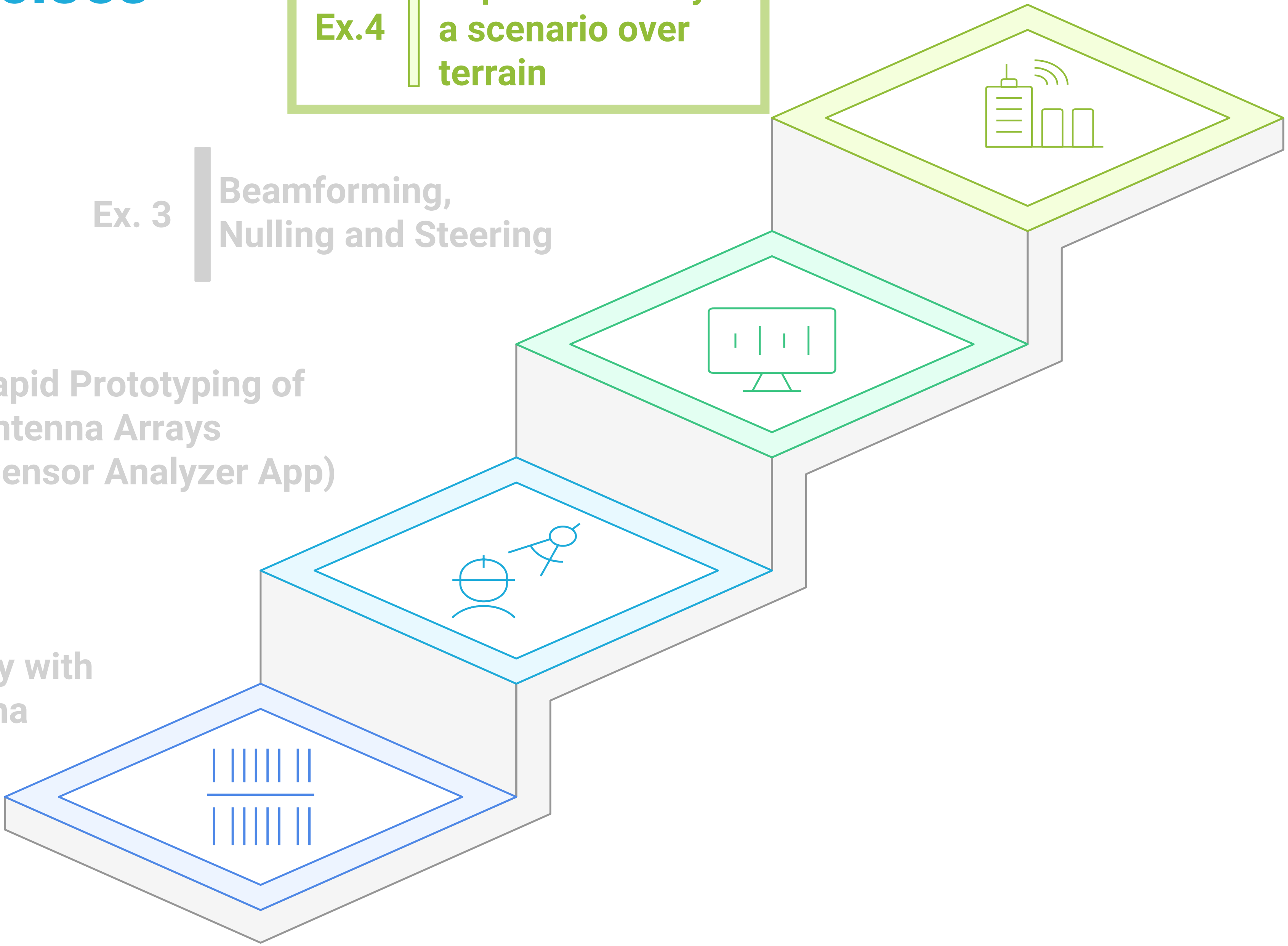
- Built foundational skills in modeling and simulating phased array systems using MATLAB
- Designed and visualized custom array configurations
- Leveraged interactive tools like the Sensor Array Analyzer for rapid design evaluation
- Applied optimization techniques to steer beams and enhance array performance
- Explored a streamlined workflow from antenna import to performance tuning

Next Steps in Your Learning Journey:

- Explore documentation and example workflows to deepen your understanding
- Take the next step with online training modules and self-paced courses on array systems, radar, and wireless
- Practice and prototype on your own. Explore bonus exercise!

Thank You!

Overview of exercises



Ex.4 Implement array in a scenario over terrain

Ex. 3 Beamforming, Nulling and Steering

Ex. 2 Rapid Prototyping of Antenna Arrays (Sensor Analyzer App)

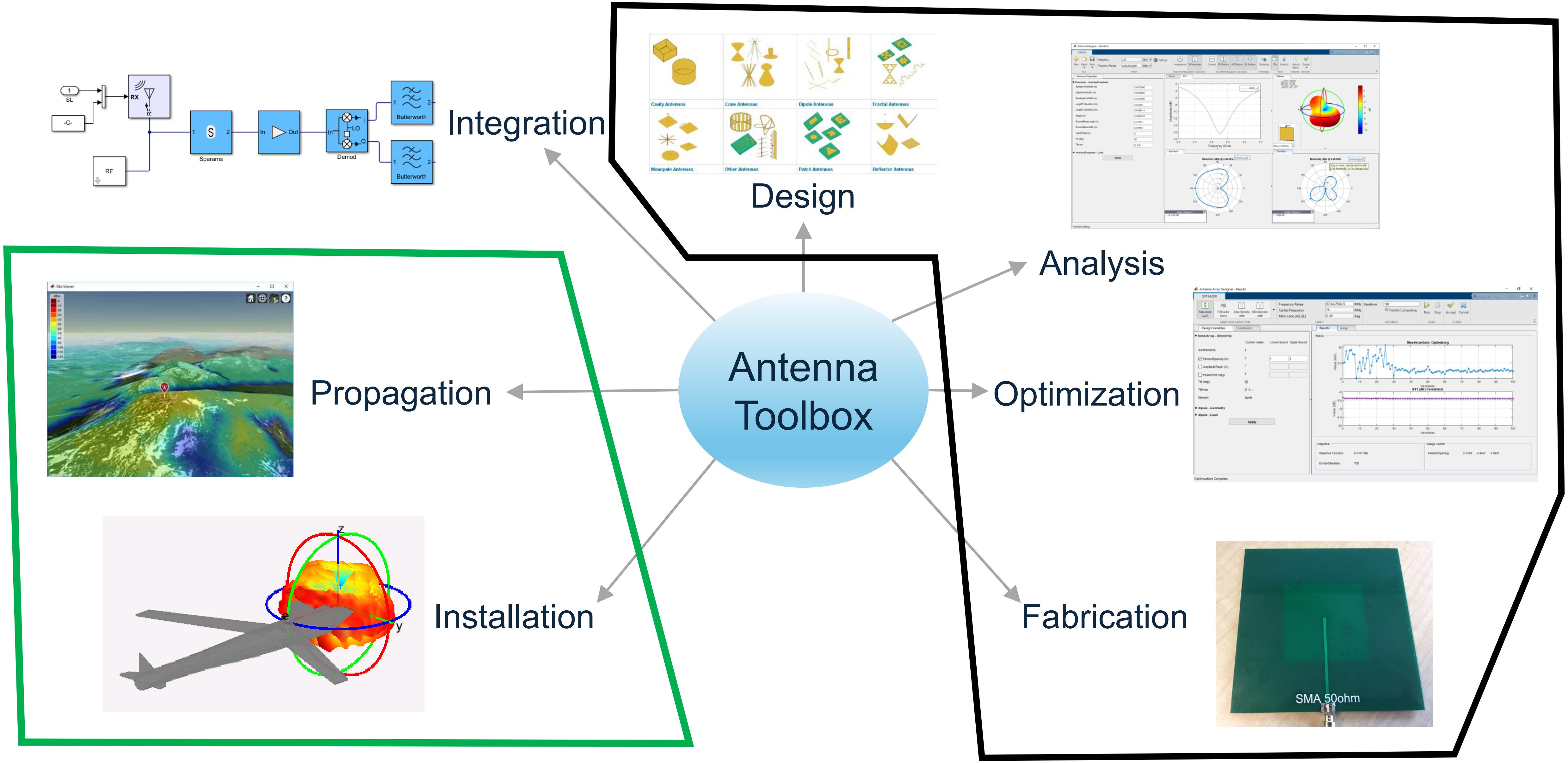
Ex. 1 Create an array with custom antenna elements

Exercise 4: Implement array in a scenario over terrain

Goals:

- Deploy antenna array in real world scenario over a terrain
- Create RF Scenario using Antenna Toolbox Siteviewer (transmit and receive sites)
- Verify Line-of-Sight Link Visibility and measure received signal and interference power levels
- Implement Interference Nulling Workflow using MVDR Beamformer

What can I do with Antenna Toolbox?



Antenna Toolbox

Integration

Design

Analysis

Optimization

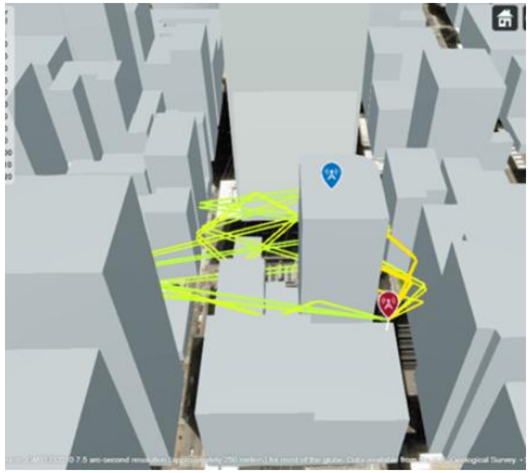
Fabrication

Installation

Propagation

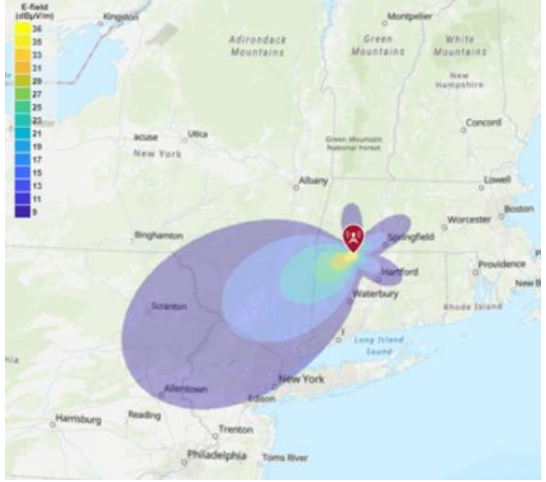
RF Propagation Models in Antenna Toolbox

- **Atmospheric Models:**
  - **Free Space:** Ideal line-of-sight (LOS) propagation.
  - **Rain, Gas, Fog:** Account for atmospheric attenuation due to weather conditions.
- **Empirical Models:**
  - **Close-In:** Statistical path loss for urban macro-cell scenarios, including non-line-of-sight (NLOS).
- **Terrain Models:**
  - **Longley-Rice (ITM) & TIREM:** Point-to-point path loss over irregular terrain, considering diffraction and ground reflection.
- **Ray Tracing Models (Advanced):**
  - **Shooting and Bouncing Rays (SBR):** Simulates reflections and diffractions (approximate path count, exact geometry).
  - **Image Method:** Exact propagation paths with precise geometric accuracy for reflections.
  - **Key Features:** Multipath analysis, material property considerations, polarization, path loss, phase shift, and GPU acceleration.

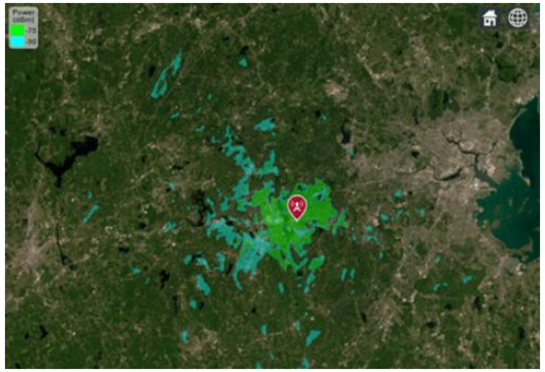


Visualization and Analysis with Antenna Toolbox

- **Interactive 3D Site Viewer:**
  - **Transmitter & Receiver Placement:** Easily define and place sites on a 3D globe or custom scene
  - **Coverage Maps:** Visualize received signal strength (e.g., dBm), SINR, or link status
  - **Propagation Path Visualization:** Plot individual ray paths showing reflections and diffractions
  - **Terrain & Basemap Integration:** Incorporate real-world terrain and basemaps for realistic context



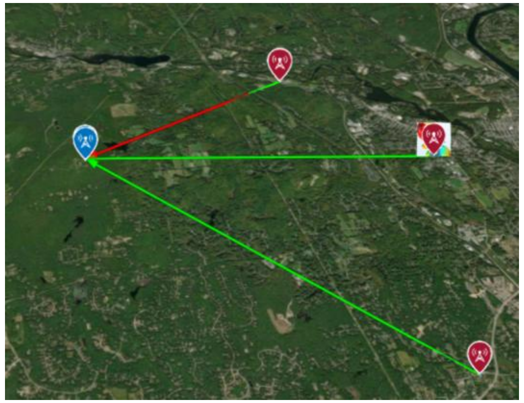
- **Analysis & Metrics:**
  - **Path Loss:** Calculate signal attenuation
  - **Signal Strength & SINR:** Determine received signal quality
  - **Link Status:** Evaluate communication link success
  - **LOS/NLOS Analysis:** Determine visibility between sites



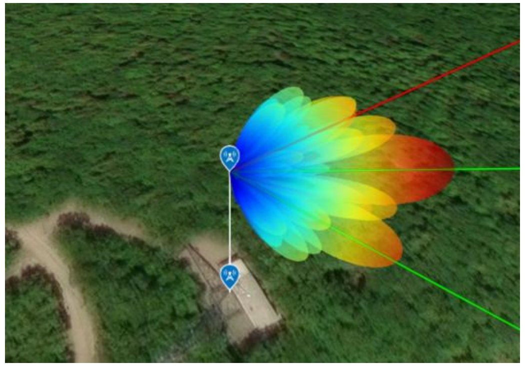
- **Integration & Customization:**
  - Define transmitter/receiver properties (txsite, rxsite).
  - Configure propagation models (propagationModel).
  - Programmatic control for automation and scripting.
  - Customize building properties from OpenStreetMap files

Exercise 4 steps: Deploy antenna array in real world scenario

- Create RF Scenario including the terrain (transmit and receive sites)
  - Create RF receive site
  - Create Tx Sites: 1 RF communication link and 2 interfering sites in 2.5 GHz Band



- Verify Line-of-Sight Link Visibility
  - Specify Receive and Transmit site antennas
  - Use / Import 8-by-8 Receive Station Antenna Array
  - Create 4-by-4 Tx Site Antenna Array



- **Calculate Received Signal Strength From Signal link and interference without Beamforming**

- **Interference Nulling Workflow using MVDR Beamformer** (next slide)

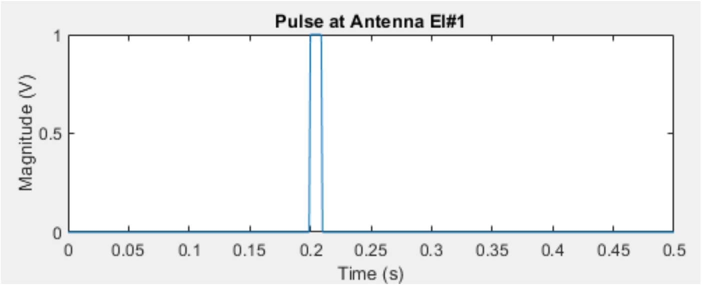
Exercise 4 steps : adaptive beamforming workflow

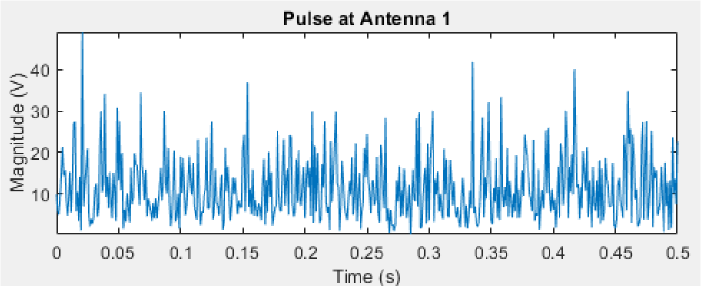
- Define Antenna Array & Signal Scenario

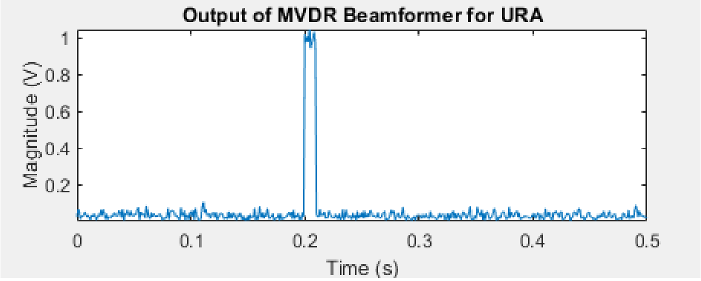
- Create Phased Array received signal
  - Generate Tx signal waveform
  - Add noise and interference signals
  - Simulate received plane waves at array (using collectPlaneWave)

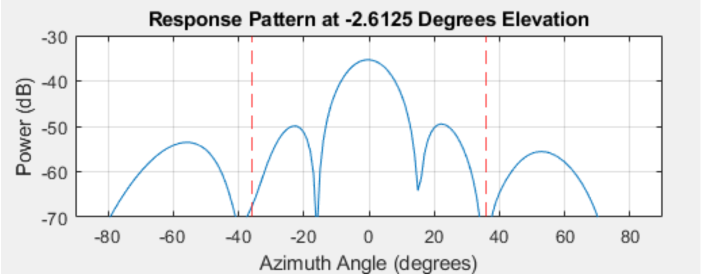
- Create MVDR Beamformer object
  - Specify URA object, input angle
- Apply MVDR Beamformer for received signal
  - Define Steering Vector for Desired Signal

- Plot Array Radiation pattern after beamforming
  - Visualize the array pattern to observe nulls in interference directions
  - compute received signal strength after beamforming









Exercise 4: functions used in exercise

RF Scenario modeling using Antenna toolbox

Siteviewer

- Display transmitter sites, receiver sites, and RF propagation visualizations by using a siteviewer object.
  - Default mode: 3-D globe — Display sites that are referenced to geographic coordinates. You can customize the globe using custom terrain, high-zoom-level or custom basemaps, and buildings.

- txsite, rxsite
  - Display transmitter and receiver sites on a 3-D globe, calculate the distance and angles between the sites, and analyze the signal strength of the transmitter at the receiver site.

- los
  - Display or compute line-of-sight (LOS) visibility status

- sigstrength (rx,tx,"freespace")

https://www.mathworks.com/help/releases/R2024b/antenna/ref/siteviewer.html

WORK ON EXERCISE 4
(20 minutes)