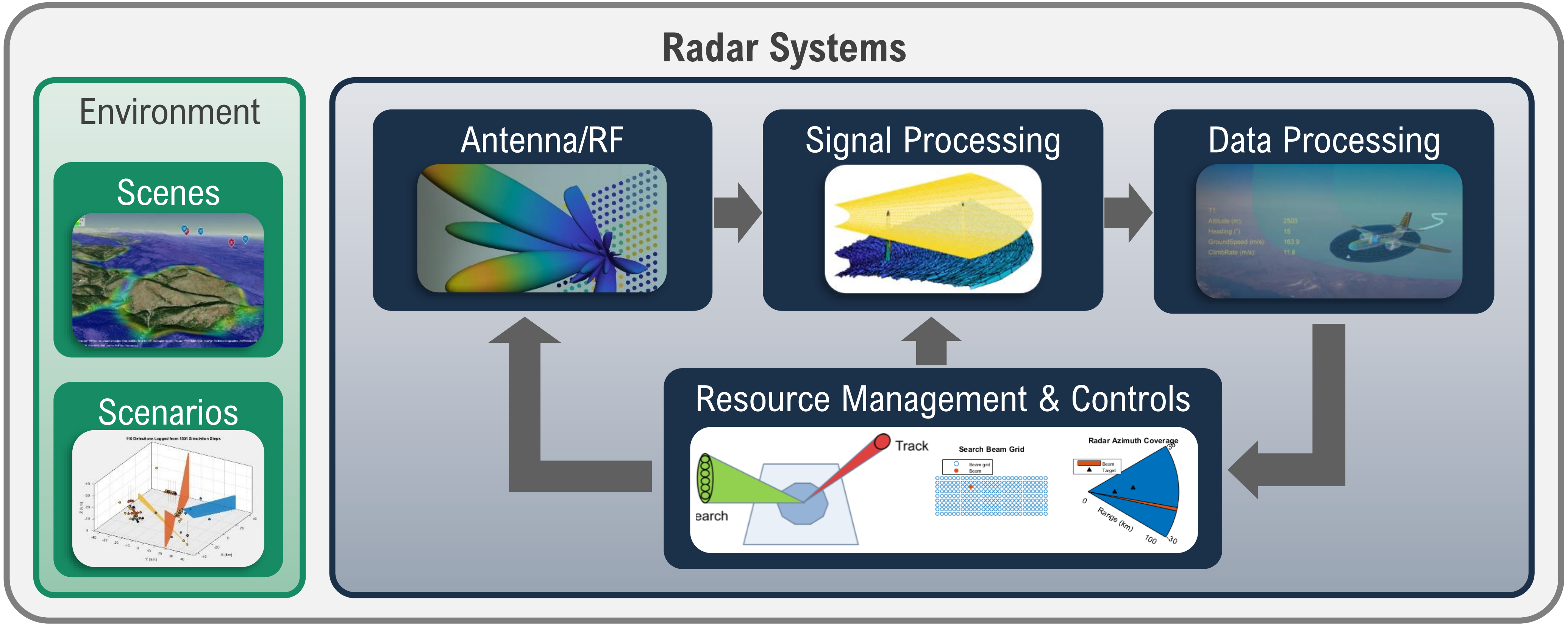
Radar System Design with MATLAB

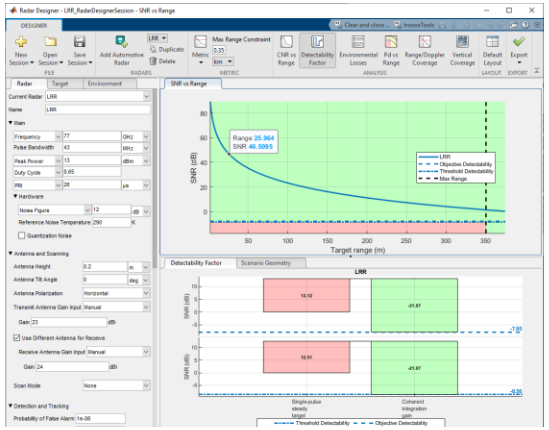
Develop Radar Systems with MATLAB

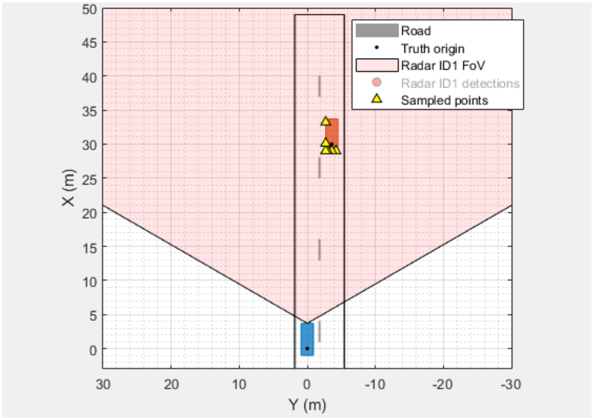


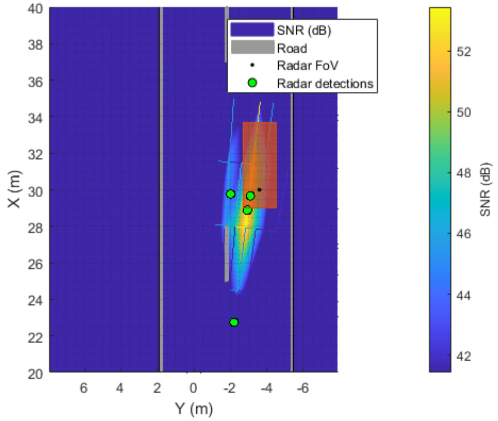
Three Abstraction Levels for Support of Full Radar Life Cycle

Less ***Fidelity*** *More*

Power-level	***Measurement-level***	***Waveform-level***
Link Budget	**Detections**	**Raw IQ**







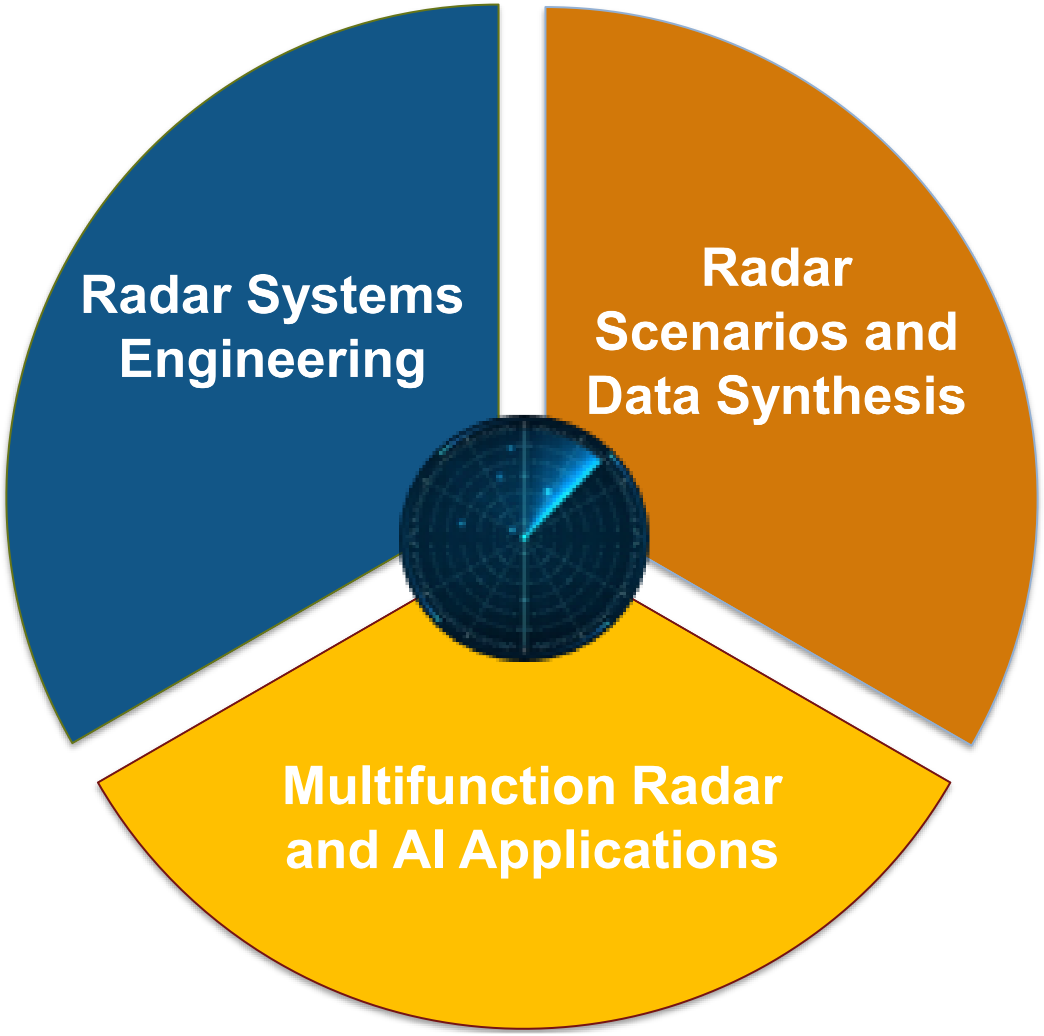
Applications: Concept Dev/Design, Systems Analysis

Applications: Systems Analysis, Scenario Analysis, Tracker Design

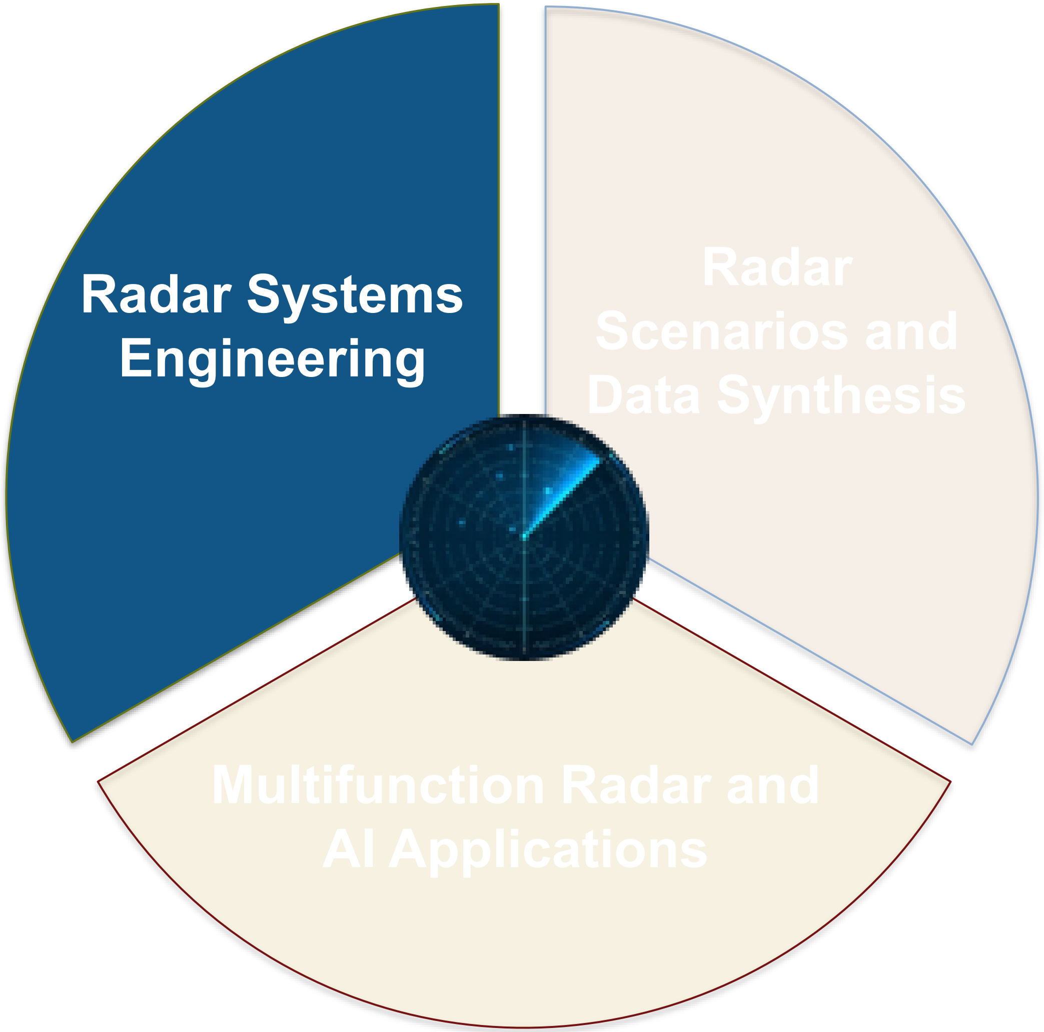
Applications: Algorithm Development, End-to-End Performance Assessment

Less ***Computational Resources*** *More*

** See Design and Simulate an FMCW Long-Range Radar (LRR) Example [here]*



- Perform link budget analysis and evaluate design trade-offs
- Simulate a radar scenario to synthesize radar data
- Example of resource management for multifunction radar
- Example using AI workflow to remove maritime clutter


Radar Systems Engineering
Radar Scenarios and Data Synthesis
Multifunction Radar and AI Applications

Define System Parameters, Target, and Environment for Link Budget Analysis

Fundamental
operating frequency, bandwidth, PRF, duty cycle, peak power

Hardware
quantization loss, beam shape loss, noise figure, noise temperature, polarization

Processing
eclipsing loss/factor, matching loss, CFAR loss MTI loss/factor, binary integration loss, STC factor

Scanning
beam-dwell factor, scan loss

Target
RCS, Swerling models, max acceleration

Surface clutter
effective Earth radius, land/sea reflectivity, roughness, permittivity, constant gamma clutter, radar propagation factor, clutter-to-noise ratio

Atmosphere
Tropospheric refraction with 6 built-in ITU models, lens loss

Precipitation
ITU fog/cloud path loss, ITU and Crane rain models, Gunn-East snow model

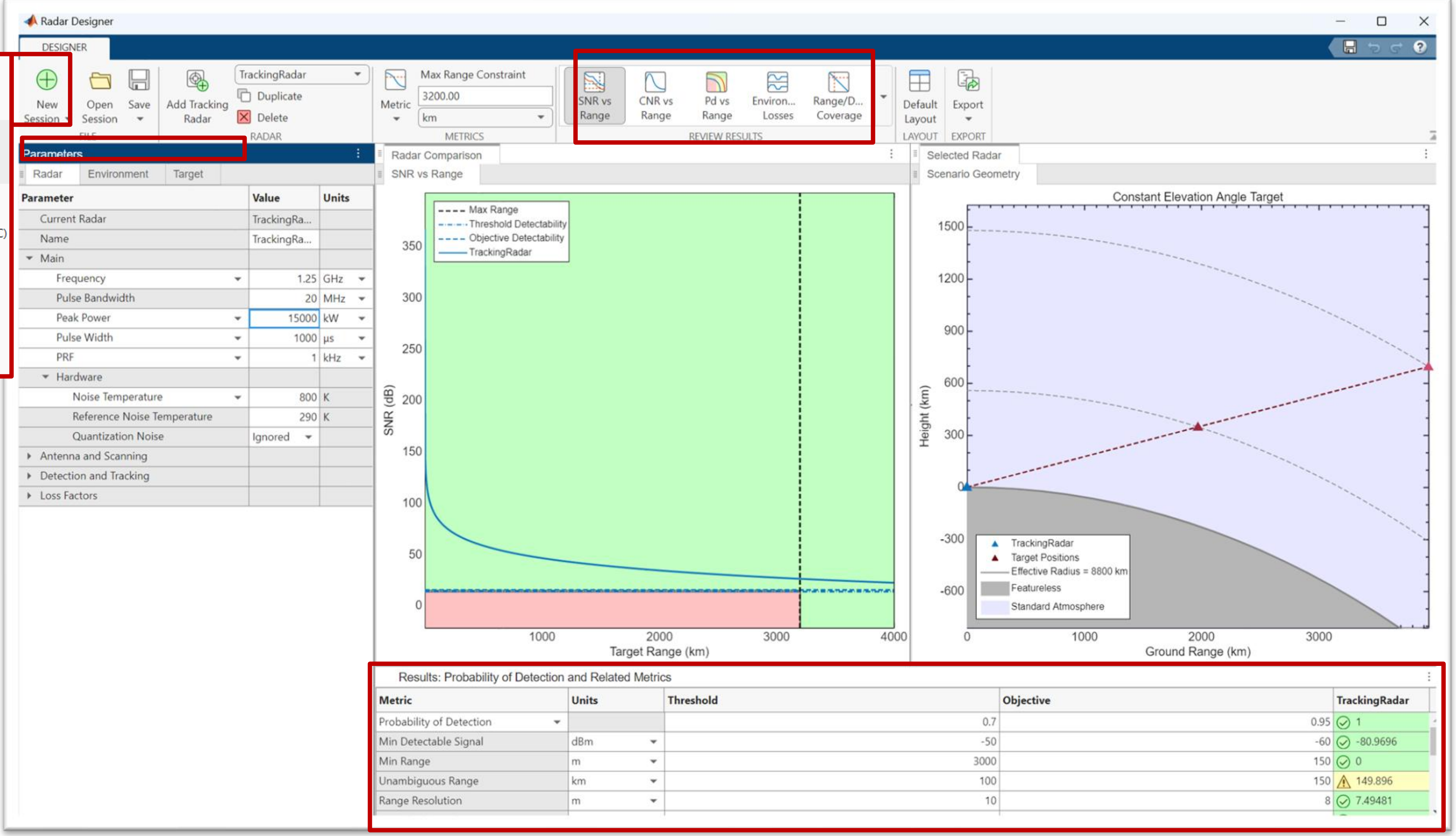
Radar Systems Engineering

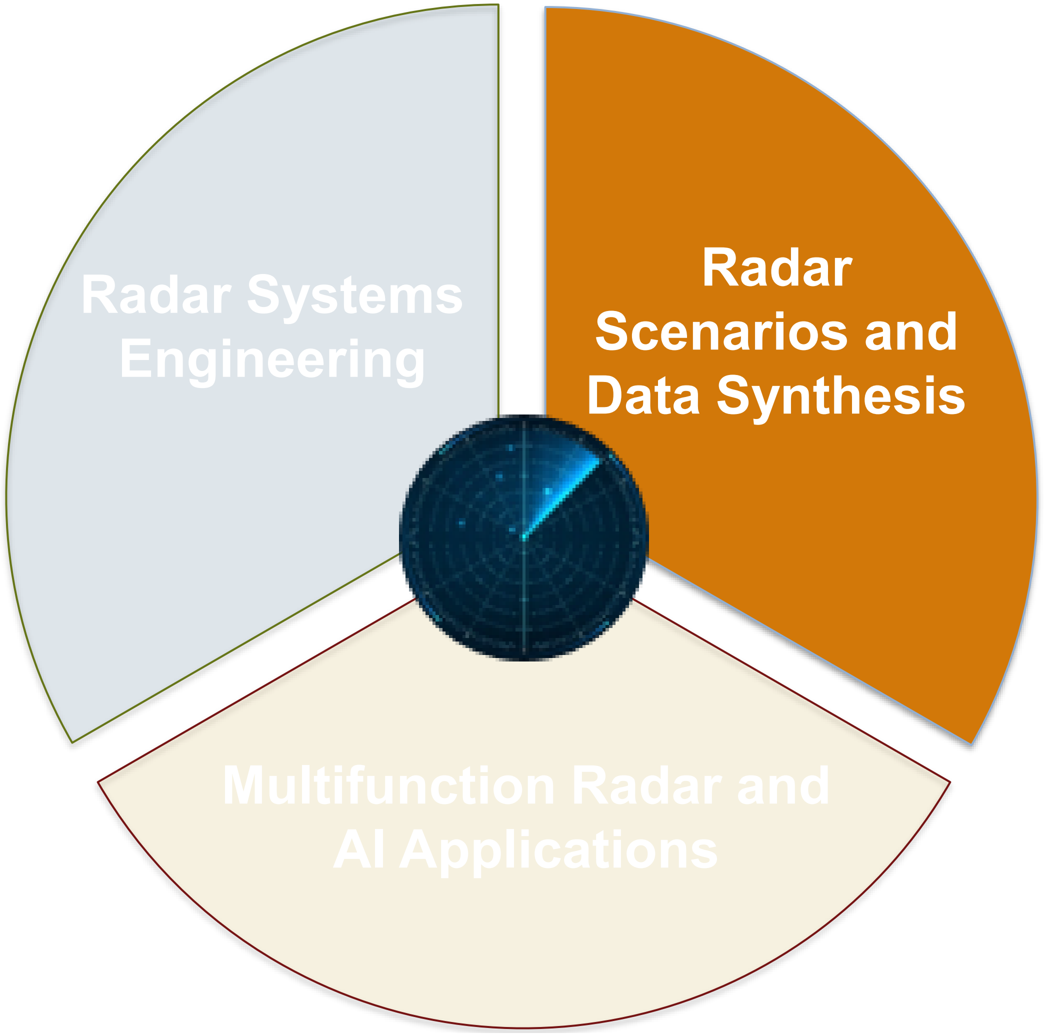
Perform Link Budget Analysis with Radar Designer App

Quick start with **5 built-in configurations**

Export MATLAB script or formatted report

- **Airborne Radar** Long-range, airborne surveillance radar
- **Airport Radar** Terminal airport surveillance radar
- **Automotive Radar** Automotive radar for Adaptive Cruise Control (ACC)
- **Tracking Radar** Ground-based, cued tracking radar system
- **Weather Radar** Clear air weather radar



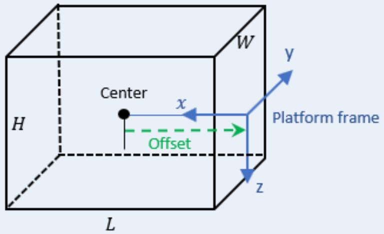

Radar Systems Engineering
Radar Scenarios and Data Synthesis
Multifunction Radar and AI Applications

Radar Scenarios and Data Synthesis

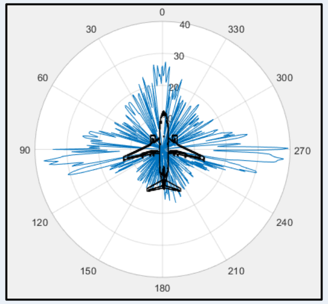
Author and Simulate Radar Scenarios

Model **Platforms** and **Targets** → Model **Environment** → Model **Trajectories** → Model **Sensors** → **Simulate** scenarios

Object Dimensions
bounding box



RCS signature
Az, el pattern
frequencies dependency



Land Surface Clutter
DTED, Custom
Land Reflectivity Models

Sea Surface Clutter
Spectral Model
Sea reflectivity models

Atmospheric Refraction
Effective Earth radius
Refractivity gradient

Use kinematic properties
acceleration, angular velocity

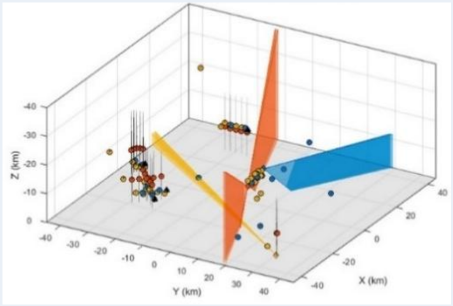
Use waypoints
position, orientation, time of arrival,
ground speed, climb rate

fixed NED or ENU frame (x,y,z) or,
geo-referenced (lat, lon, alt)

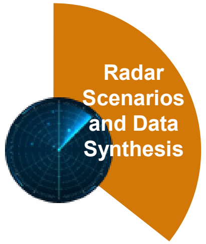
Measurement Level
Scenario Analysis,
Tracker Design

Waveform Level
Algorithm Development,
End-to-End Performance Assessment

Generate radar data
I/Q signals, detections, tracks



Monte Carlo
perturb ground truth and sensor to increase testing robustness

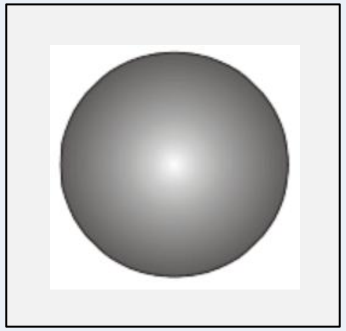


Model Radar Targets with Different Levels of Fidelity

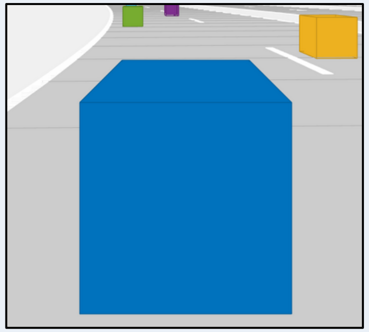
Low — Modeling Complexity — High

Model Platforms and Targets | Model Environment | Model Trajectories | Model Sensors | Simulate scenarios

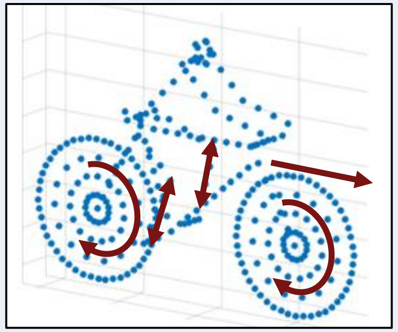
Monostatic & Bistatic Signature Modeling



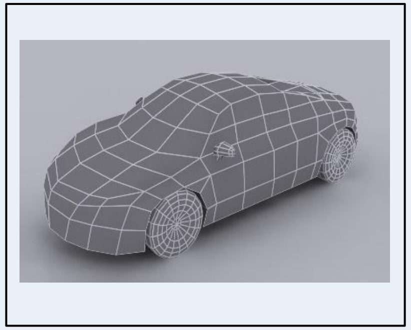
Point



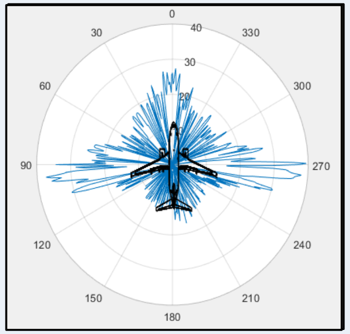
Cuboid
(Multi-scatterer)



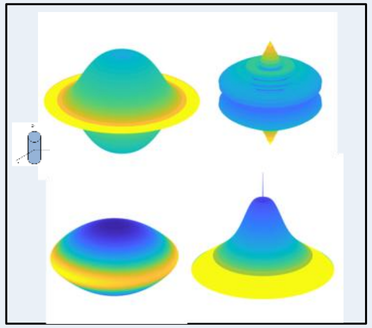
Multi-scatterer
(non-rigid body)



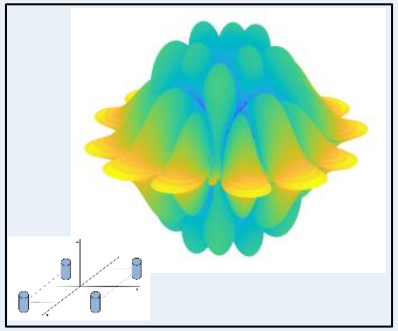
Mesh



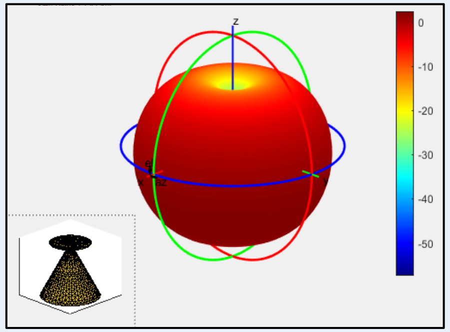
RCS Pattern



Analytic results
(for basic shapes)



Superposition
(of basic shapes)



EM Solver

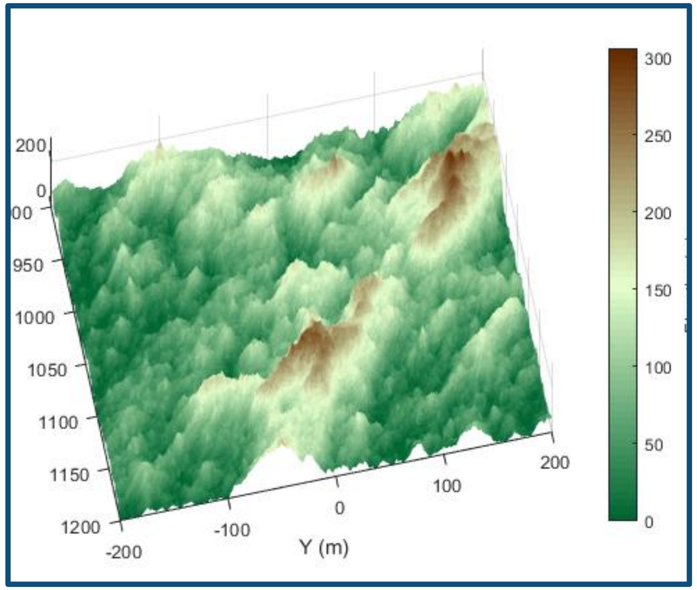
Link to Example

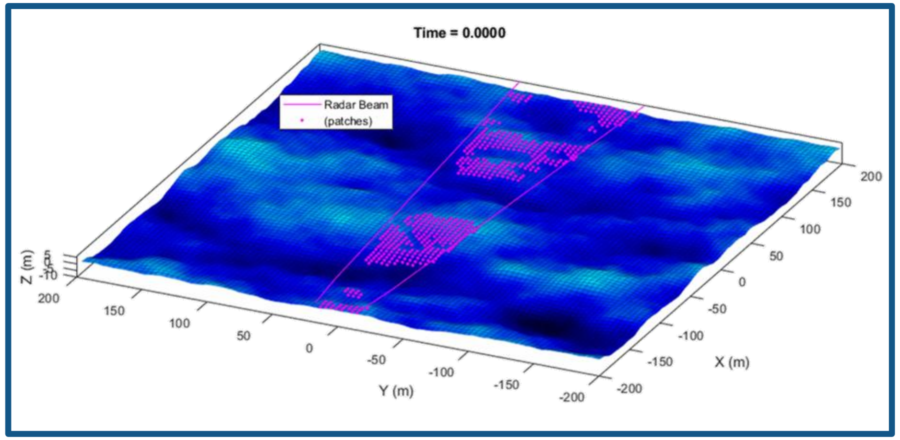
Radar Scenarios and Data Synthesis

Increase the Fidelity by Modeling the Environment

Model Platforms and Targets | **Model Environment** | Model Trajectories | Model Sensors | Simulate scenarios

Land Surface Clutter





Sea Surface Clutter

Atmospheric Refraction

Link to Example

Model Land Surfaces with Elevation Height Map and Surface Reflectivity

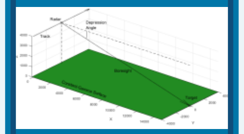
Radar Scenarios and Data Synthesis

Model Platforms and Targets | **Model Environment** | Model Trajectories | Model Sensors | Simulate scenarios

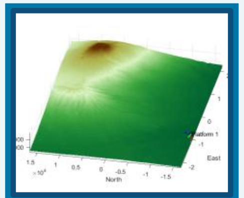
Land Surface Clutter

Sea Surface Clutter

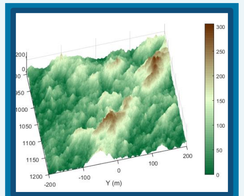
Atmospheric Refraction



Flat



DTED



Custom

Model	Type	Land Types	Grazing Angle			Frequency (GHz)
			Low	Mid	High	
APL (ADSAM)	Mathematical	*Urban, High-relief, Low-relief*	✓	✓	✓	1 – 100
Barton*	Mathematical	*Rugged Mountains, Mountains, Metropolitan, Urban, Wooded Hills, Rolling Hills, Woods, Farm, Desert, Flatland, Smooth*	×	✓	×	1 - 10
Billingsley	Empirical	*Low-relief Rural, Low-relief Forest, Farm, Desert, Marsh, Grassland, High-relief Rural, High-relief Forest, Mountains, Urban, Low-relief Urban*	✓	×	×	0.003 - 12
GIT	Semi-Empirical	*Soil, Grass, Tall Grass, Trees, Urban*	×	✓	×	3 - 15
Morchin	Mathematical	*Desert, Farm, Woods, Mountains*	×	×	✓	0.3 - 8
Nathanson	Empirical	*Desert, Farm, Woods, Jungle, Rolling Hills, Urban*	✓	✓	×	1 - 36
Ulaby Dobson	Semi-Empirical	*Soil, Grass, Shrubs, Short Vegetation*	✓	✓	×	1 - 18

Legend	
✓	Model Valid
×	Model Not Valid

Abbreviations

APL – Applied Physics Laboratory ADSAM - Air-Directed Surface-to-Air Missile GIT – Georgia Institute of Technology

Built-in ground surface reflectivity models + bring your own

Link to Example

Radar Scenarios and Data Synthesis

Model Sea Surfaces with Motion

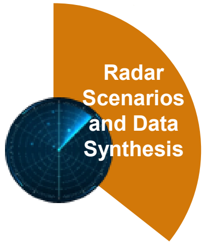
Model Platforms and Targets | **Model Environment** | Model Trajectories | Model Sensors | Simulate scenarios

- Land Surface Clutter
- **Sea Surface Clutter**
- Atmospheric Refraction

Sea Motion Model = Spectral Model × Spreading Function

Wind Direction | Wind Speed | Fetch

[Link to Example]

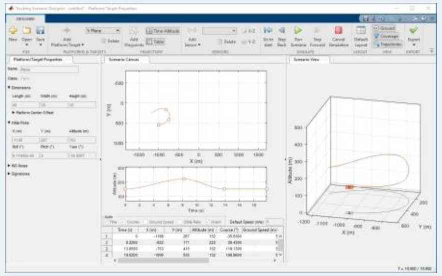


Model trajectories for platforms and targets

Model Platforms and Targets | Model Environment | **Model Trajectories** | Model Sensors | Simulate scenarios

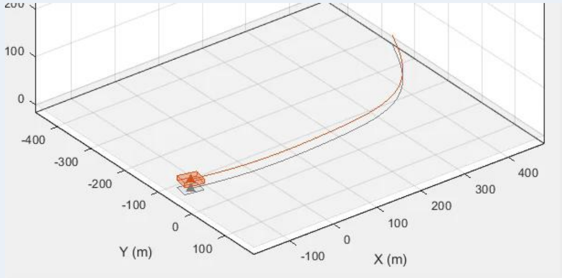
Automatic banking (roll)

lock pitch to angle-of-attack (fixed wing aircraft)

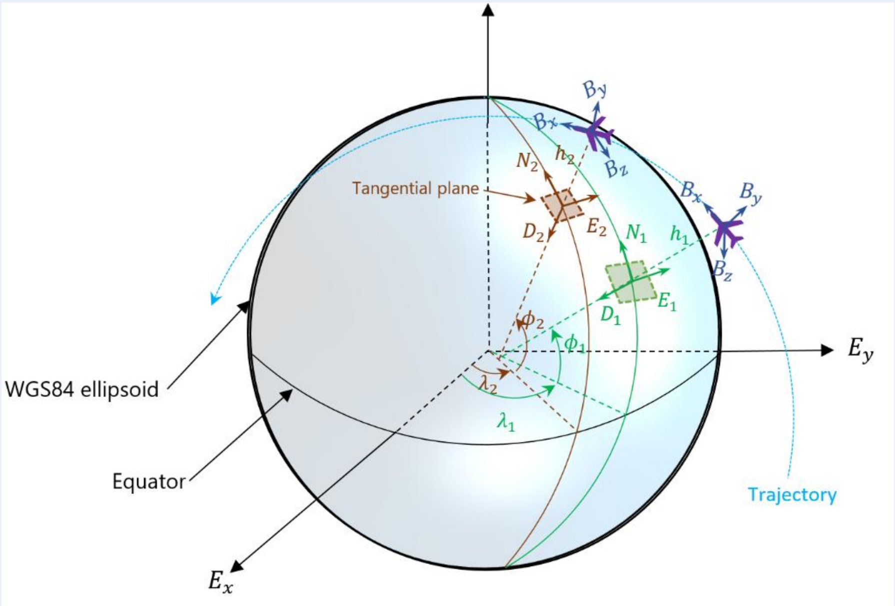


Automatic pitch and roll

to conserve momentum (quadcopter/rotary wing)



ECEF and Local Frames



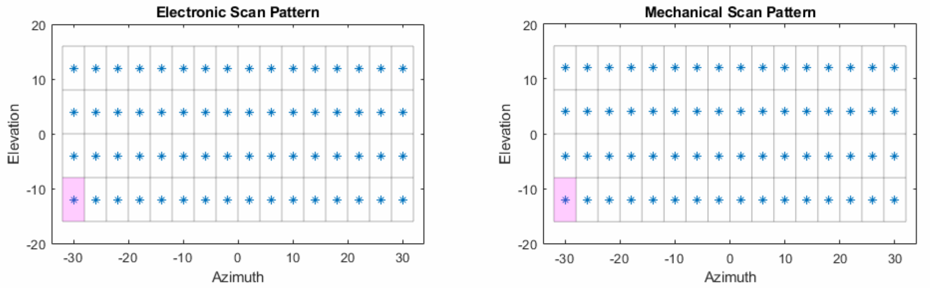
Link to Example

Radar Scenarios and Data Synthesis

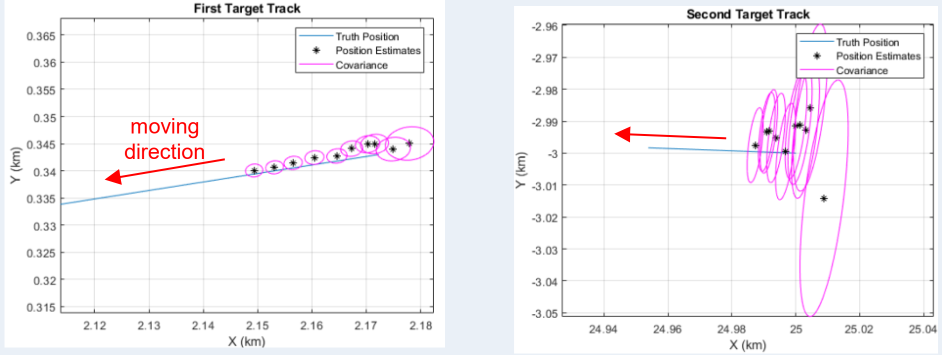
Simulate detections, clusters and tracks

Model Platforms and Targets | Model Environment | Model Trajectories | **Model Sensors** | Simulate scenarios

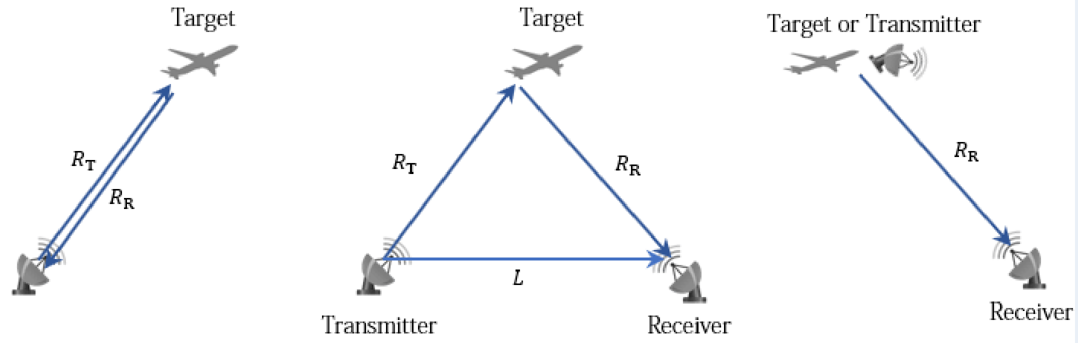
Scan Patterns



Fast emulation of tracks and track covariances

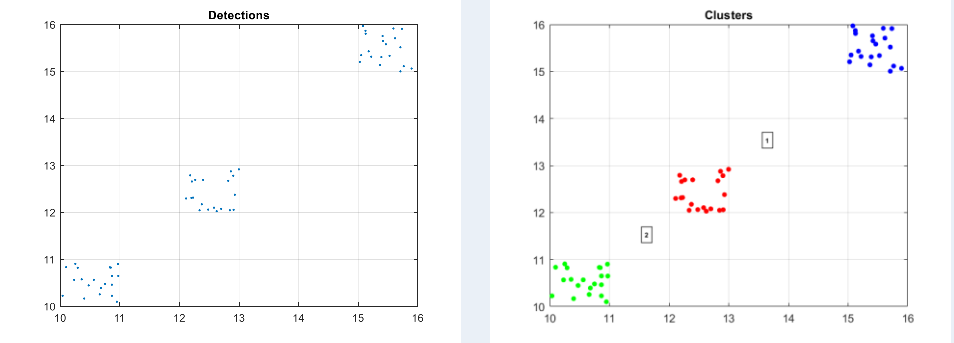


Detection Modes



(a) Monostatic (b) Bistatic (c) ESM

Clustering



Measurement-level

[Link to Example](#)

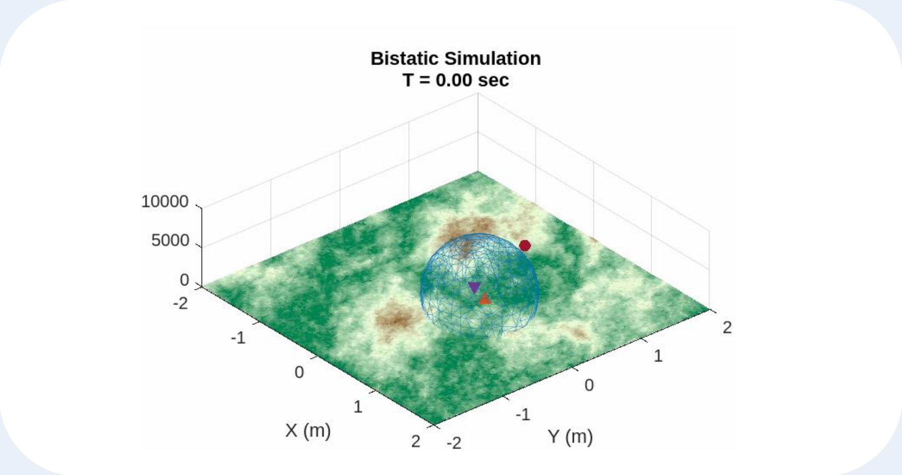
Radar Scenarios and Data Synthesis

Simulate I/Q signals

Model Platforms and Targets | Model Environment | Model Trajectories | **Model Sensors** | Simulate scenarios

Physics-based Monostatic and Bistatic Models*

- Waveform
- Antenna
- Polarization
- Frequency
- Tx gain
- Rx gain
- Peak power
- Noise figure
- Interference*
- etc…



`bistaticTransmitter + bistaticReceiver`

Simulate IQ from multiple transmitters and receivers

- Simulate in bistatic pairs
- Supports CPU parallelization
- Enables usage of user-provided propagation models
- Can work independently or in conjunction with `radarScenario`

***New in R2025a!**

Waveform-level

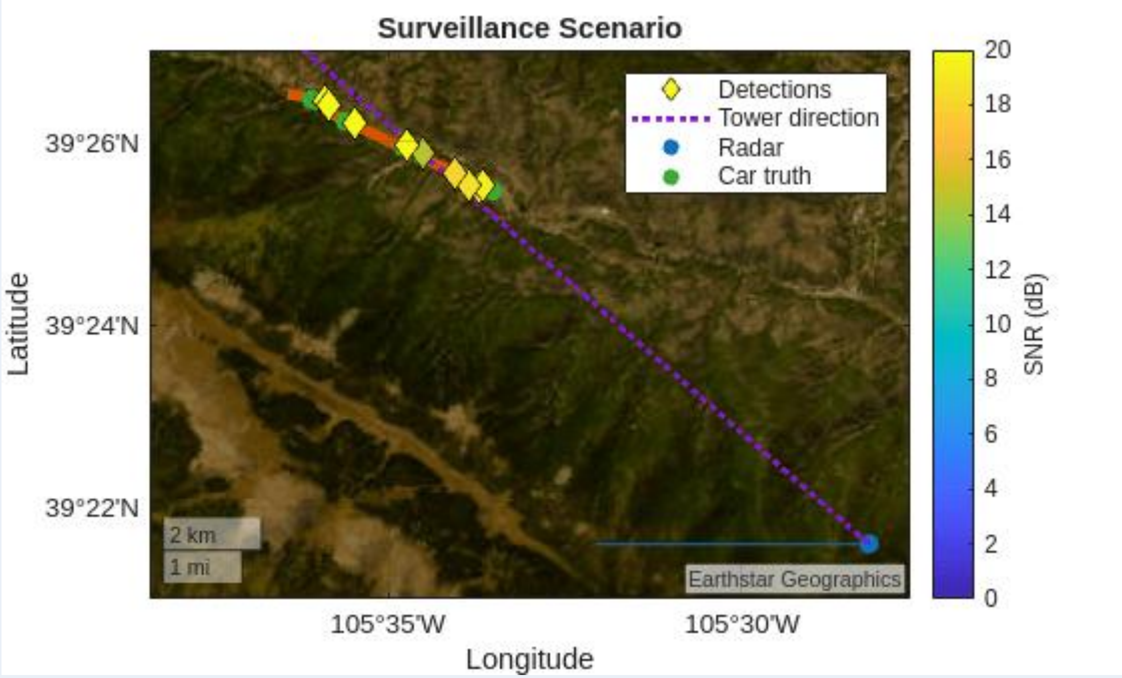
Radar Scenarios and Data Synthesis

Demo: Parallel Simulation of Target, Clutter, and Interference Signals

Model **Platforms** and **Targets** → Model **Environment** → Model **Trajectories** → Model **Sensors** → **Simulate** scenarios

Parallel Simulation of Target, Clutter, and Interference Signals

- Create a physics-based, time-evolving scenario at the waveform level incorporating clutter, interference, and targets
- Parallelize the simulation for enhanced clutter return generation speed
- Design processing to mitigate interference and clutter and detect target returns



Waveform-level

Model Radar Targets with Different Levels of Fidelity

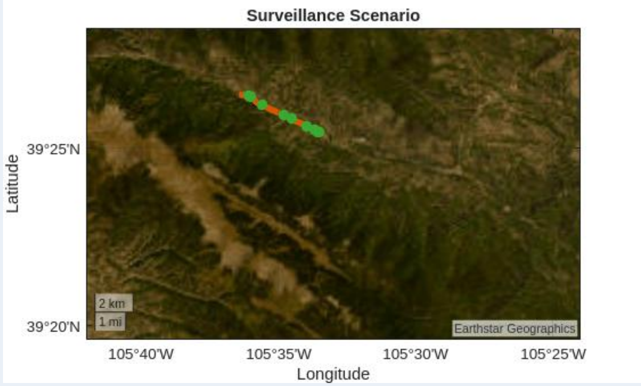
Radar Scenarios and Data Synthesis

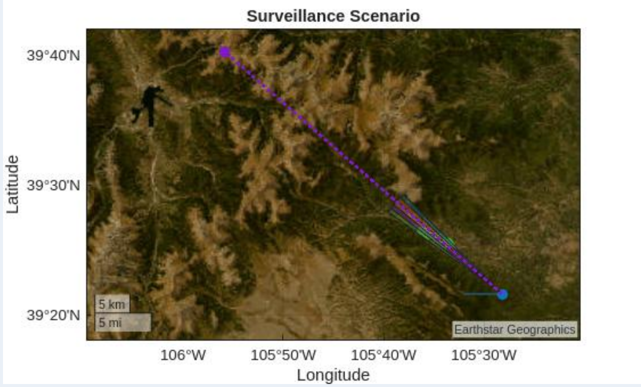
Low — Modeling Complexity — High

Model Platforms and Targets | Model Environment | Model Trajectories | Model Sensors | Simulate scenarios

Define Target and Interference Platforms

- Create car targets positioned along a road
- Define a constant point RCS for each of the cars
- Add an interferer to the scenario, positioned at the purple point





Waveform-level

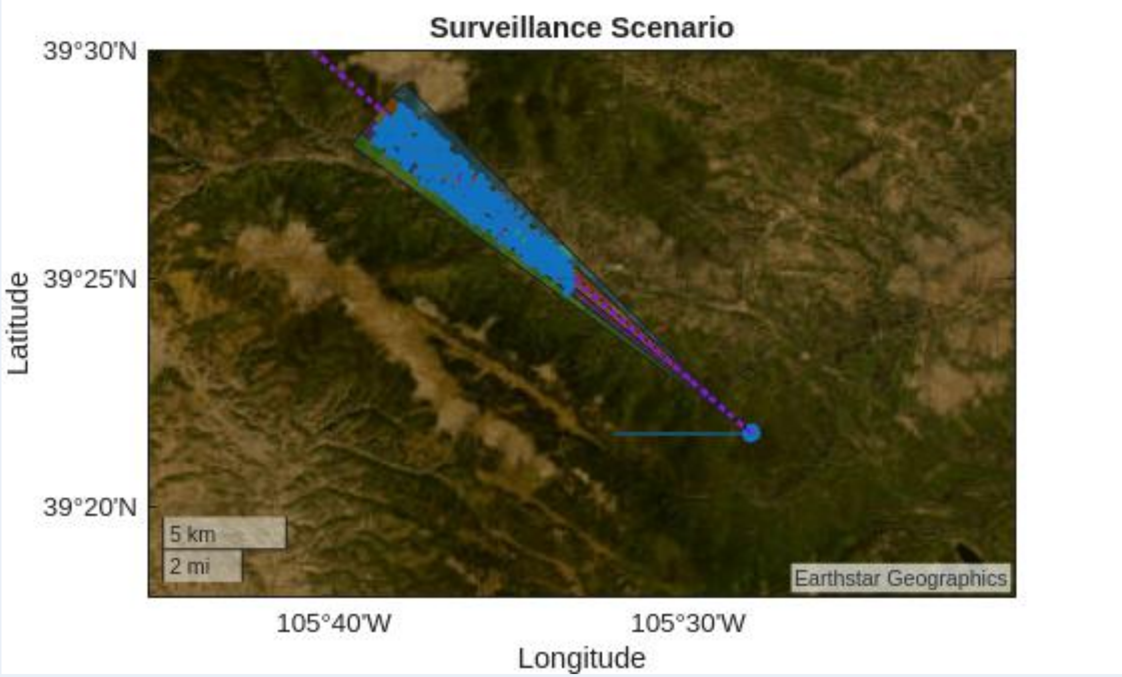
Increase the Fidelity by Modeling the Environment

Radar Scenarios and Data Synthesis

Model Platforms and Targets | **Model Environment** | Model Trajectories | Model Sensors | Simulate scenarios

Define Environmental Clutter

- Add a land surface and clutter model to the road
- Clutter targets are visualized as the blue points



Waveform-level

Model trajectories for platforms and targets

Radar Scenarios and Data Synthesis

Model Platforms and Targets | Model Environment | **Model Trajectories** | Model Sensors | Simulate scenarios

Define Motion

- Define trajectories using an LLA-based waypoint trajectory and typical motor vehicle speeds along the road for the cars

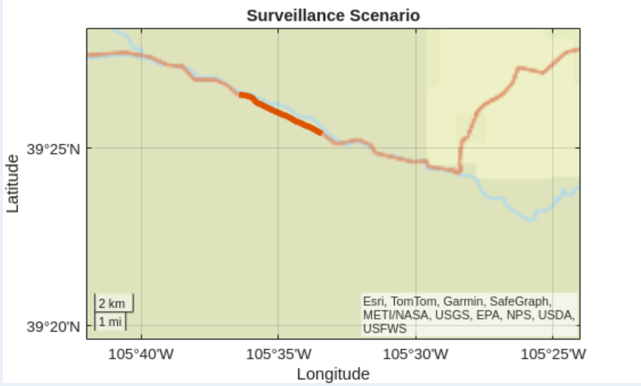
```
% Interpolate points along the road to define the car's trajectory
ds = spds(m)*toas + d0(m);
wpts = interp1(dist,enu,ds);
[lts,lns,hts] = enu2geodetic(wpts(:,1),wpts(:,2),wpts(:,3),wptsRoad(1,1),wptsRoad(2,1),wptsRoad(3,1),earth);
thisCar = platform(scenario,Trajectory=geoTrajectory([lts lns hts],toas));
```

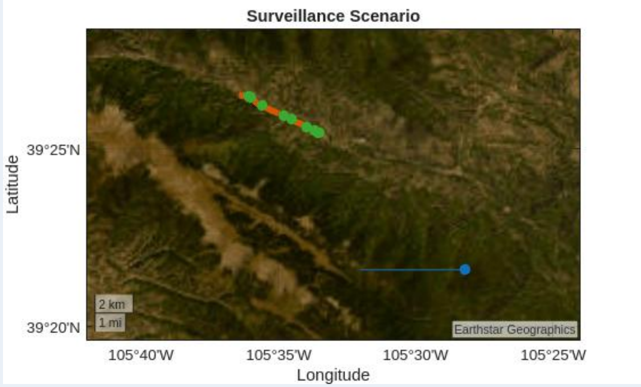
- Define the radar platform's trajectory as moving is a straight line to the west

```
rdrLLA = [39.36 -105.47 7e3]; % [deg deg m]
rdrHeading = 270; % deg, West
rdrSpeed = 180;   % Speed in knots
rdrDur = 60;      % Duration in seconds
[wpts,toa] = helperConstantVelocityGeoWaypoints(rdrLLA,rdrHeading,rdrSpeed,rdrDur);
```

Add a radar platform to the scenario that uses a `geoTrajectory` to traverse these waypoints.

```
rdrPlat = platform(scenario,Trajectory= ...
    geoTrajectory(wpts,toa,ReferenceFrame="ENU"));
```





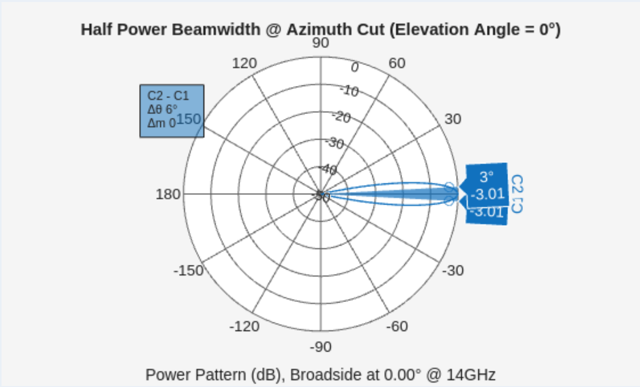
Waveform-level

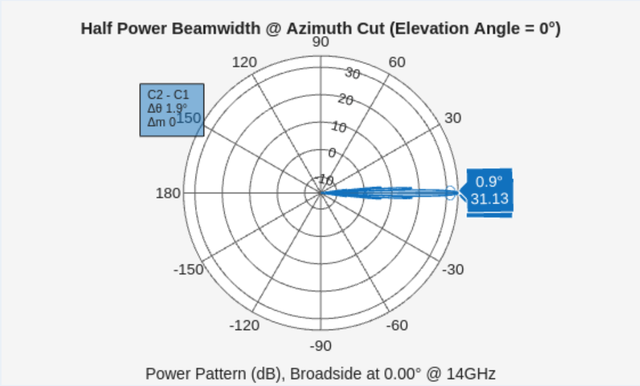
Define Monostatic Radar with New Bistatic Components

Model Platforms and Targets | Model Environment | Model Trajectories | **Model Sensors** | Simulate scenarios

Model Monostatic Radar Sensor

- Define the radar including:
  - Transmit Antenna (bistaticTransmitter)
    - Wide beamwidth for swath illumination (6° 3dB beamwidth)
  - Receive Array (bistaticReceiver)
    - Narrow beamwidth for better receive resolution (2° 3dB beamwidth)
  - Waveform
    - LFM waveform
  - Receiver





Define Interferer with New Bistatic Transmitter

Radar Scenarios and Data Synthesis

Model Platforms and Targets | Model Environment | Model Trajectories | Model **Sensors** | Simulate scenarios

Model Interferer

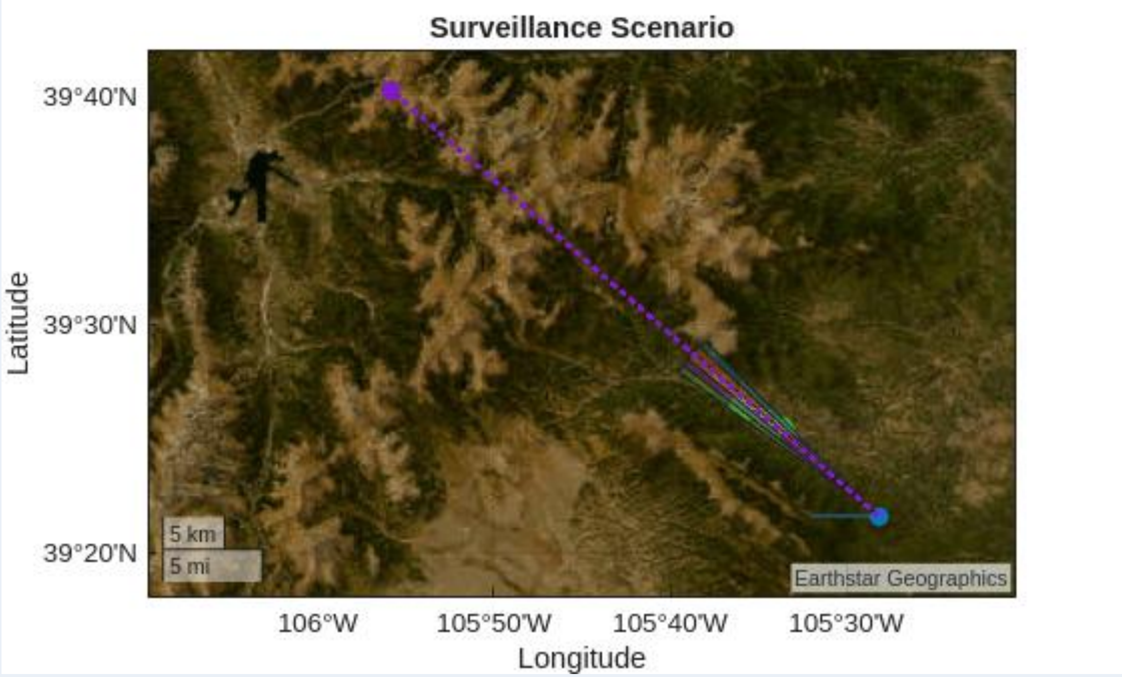
- Define an interferer using a second bistaticTransmitter emitting a phase coded waveform

```
twrTx = bistaticTransmitter();
wfm = phased.PhaseCodedWaveform( ...
    SampleRate=Fs,ChipWidth=1/Fs,PRF=rdrTx.Waveform.PRF);
wfm.NumChips = floor(sqrt(1./(wfm.ChipWidth*wfm.PRF)))^2;
twrTx.Waveform = wfm;

twrTx.TransmitAntenna.Sensor = phased.IsotropicAntennaElement;
twrTx.TransmitAntenna.OperatingFrequency = freq;
disp(twrTx)
```

```
  bistaticTransmitter with properties:

           Waveform: [1×1 phased.PhaseCodedWaveform]
        Transmitter: [1×1 phased.Transmitter]
    TransmitAntenna: [1×1 phased.Radiator]
     IsTransmitting: 0
        InitialTime: 0
     SimulationTime: 0
```



Waveform-level

Simulate I/Q Returns

Model Platforms and Targets | Model Environment | Model Trajectories | Model Sensors | **Simulate scenarios**

Simulate Scenario

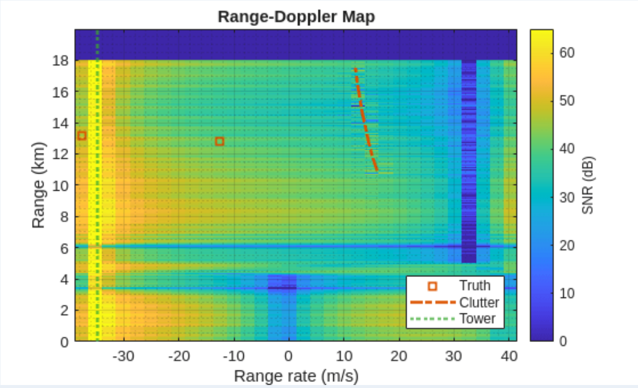
- Use a Scenario Loop to simulate one CPI
  - The scenario will step forward one time step during each loop iteration.

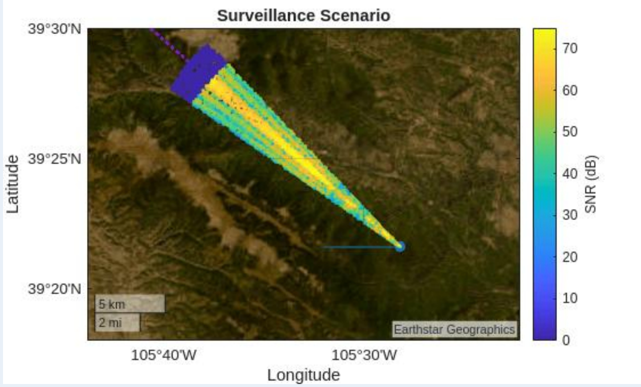
```
isRunning = advance(scenario);
while isRunning && ~isCPIDone
    simTime = scenario.SimulationTime;
```

  - Transmit, collect, and receive I/Q

```
[txSigs,txInfo] = transmit(thisTx.Transmitter,propPaths,simTime);
propSigs = propSigs + collect(rdrRx,txSigs,txInfo,propPaths);
```

  - Parallelize clutter simulation for enhanced speed
  - Beamform and Visualize Beamformed Data





Waveform-level

Radar Scenarios and Data Synthesis

Process Received Returns

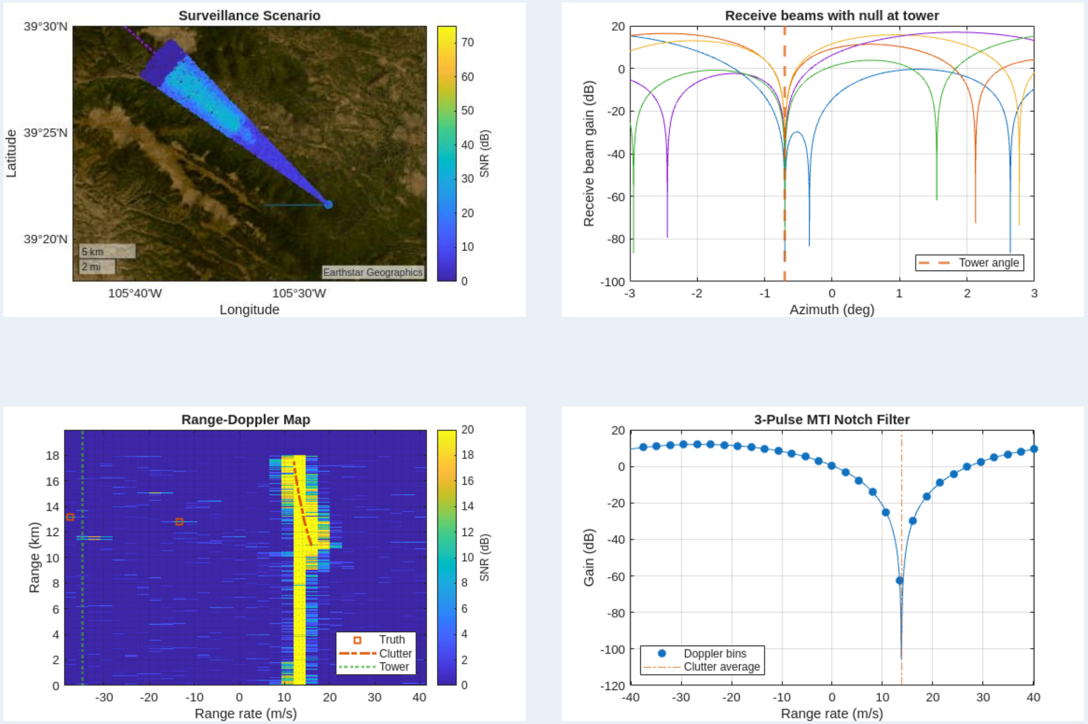
Model Platforms and Targets | Model Environment | Model Trajectories | Model Sensors | **Simulate scenarios**

Mitigate Interference and Clutter

- Use MVDR to place a null in the direction of the interferer

```
swScan = mvdrweights(getElementPosition(rdrRx.ReceiveAntenna.Sensor)/lambda,azScan,Sn);
```

- Use an MTI filter to mitigate clutter



Waveform-level

Generate Detections

Radar Scenarios and Data Synthesis

Model Platforms and Targets | Model Environment | Model Trajectories | Model Sensors | **Simulate scenarios**

Detection and Clustering

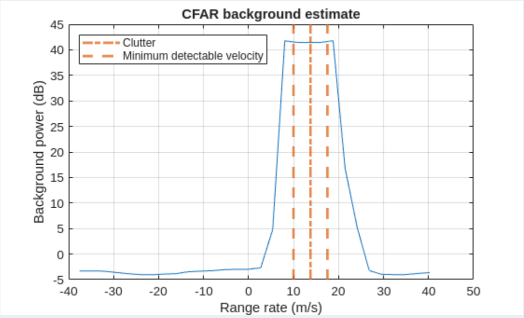
- Use CFAR to generate detections

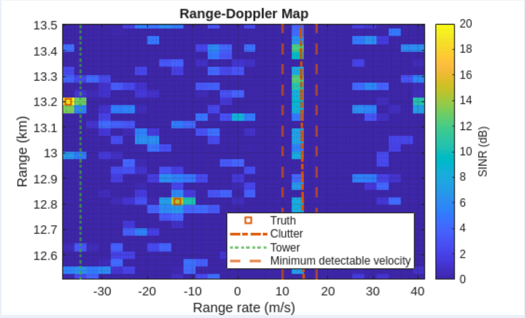
```
cfar = phased.CFARDetector2D( ...
     GuardBandSize=[1 1], ...
     TrainingBandSize=[20 1], ...
     ProbabilityFalseAlarm=1e-6, ...
     NoisePowerOutputPort=true);
```

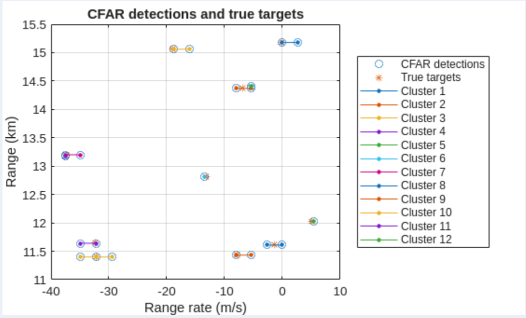
- Cluster raw detections with DBSCAN

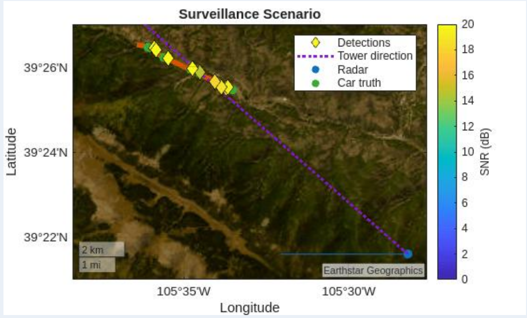
```
cluster = clusterDBSCAN(Epsilon=epsilon,MinNumPoints=2, ...
     EnableDisambiguation=true,AmbiguousDimension=2);
```

- Plot final detections
    - Options for extension: add tracking!







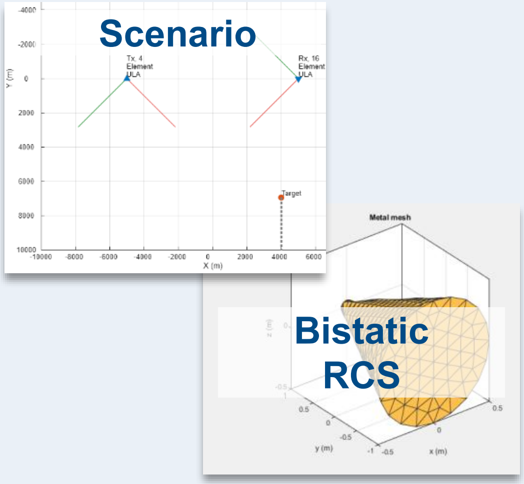


Waveform-level

Cooperative Bistatic Radar IQ Sim and Processing

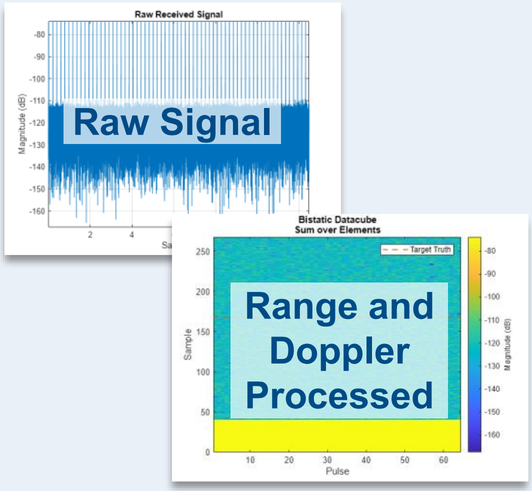
Setup Bistatic Scenario → **Simulate Bistatic Datacube** → **Remove Direct Path Return** → **Obtain Bistatic Detections**

Setup Bistatic Scenario



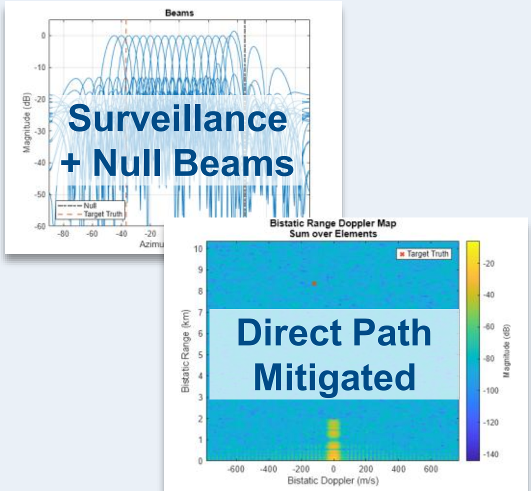
- Configure scenario with single transmitter, receiver, and target
- Calculate BRCS using Antenna Toolbox

Simulate Bistatic Datacube



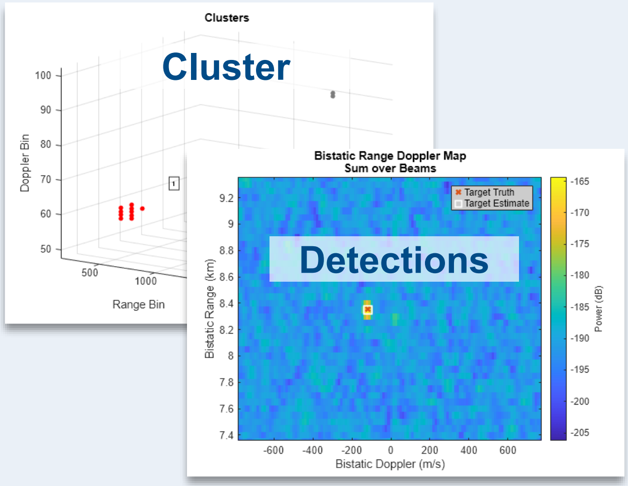
- Update paths with BRCS
- Simulate raw IQ signals
- Form datacube
- Perform range and Doppler processing

Remove Direct Path Return



- Form surveillance beams for targets
- Form null beam to mitigate direct path

Obtain Bistatic Detections



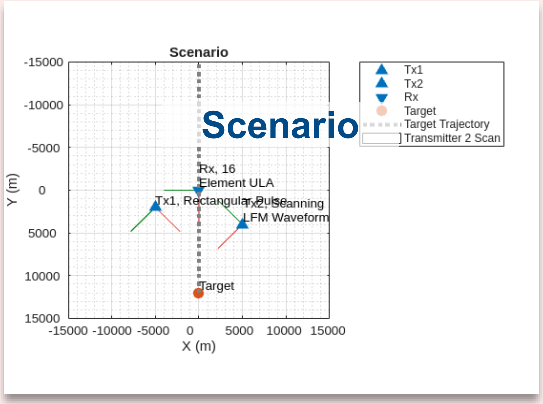
- Run CFAR
- Cluster with DBSCAN
- Perform parameter estimation

Demonstrates IQ simulation for a cooperative bistatic system with realistic signal processing to create bistatic detections

** BBRCS – Bistatic Radar Cross Section   CFAR – Constant False Alarm Rate   DBSCAN – Density-Based Spatial Clustering of Applications with Noise*

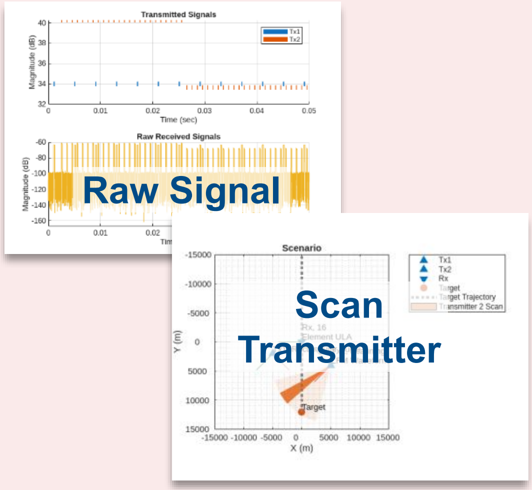
Non-Cooperative Bistatic Radar IQ Simulation and Processing

Setup Passive Bistatic Scenario



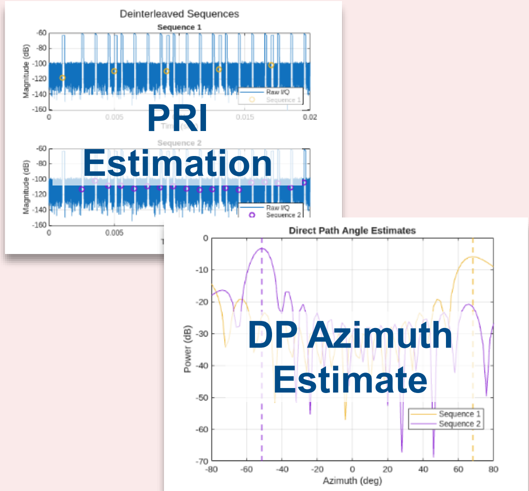
- Configure multistatic scenario
  - 2 transmitters
  - 1 receiver
  - 1 target

Simulate Passive Receiver



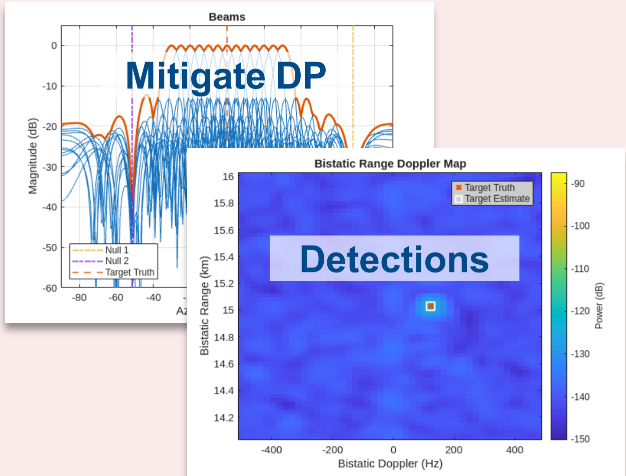
- Discuss MBET
- Manually scan a bistatic transmitter
- Simulate raw IQ signal at a passive receiver

Create Bistatic Datacube



- Deinterleave
- Identify reference waveform
- Estimate direct path (DP) azimuth angle

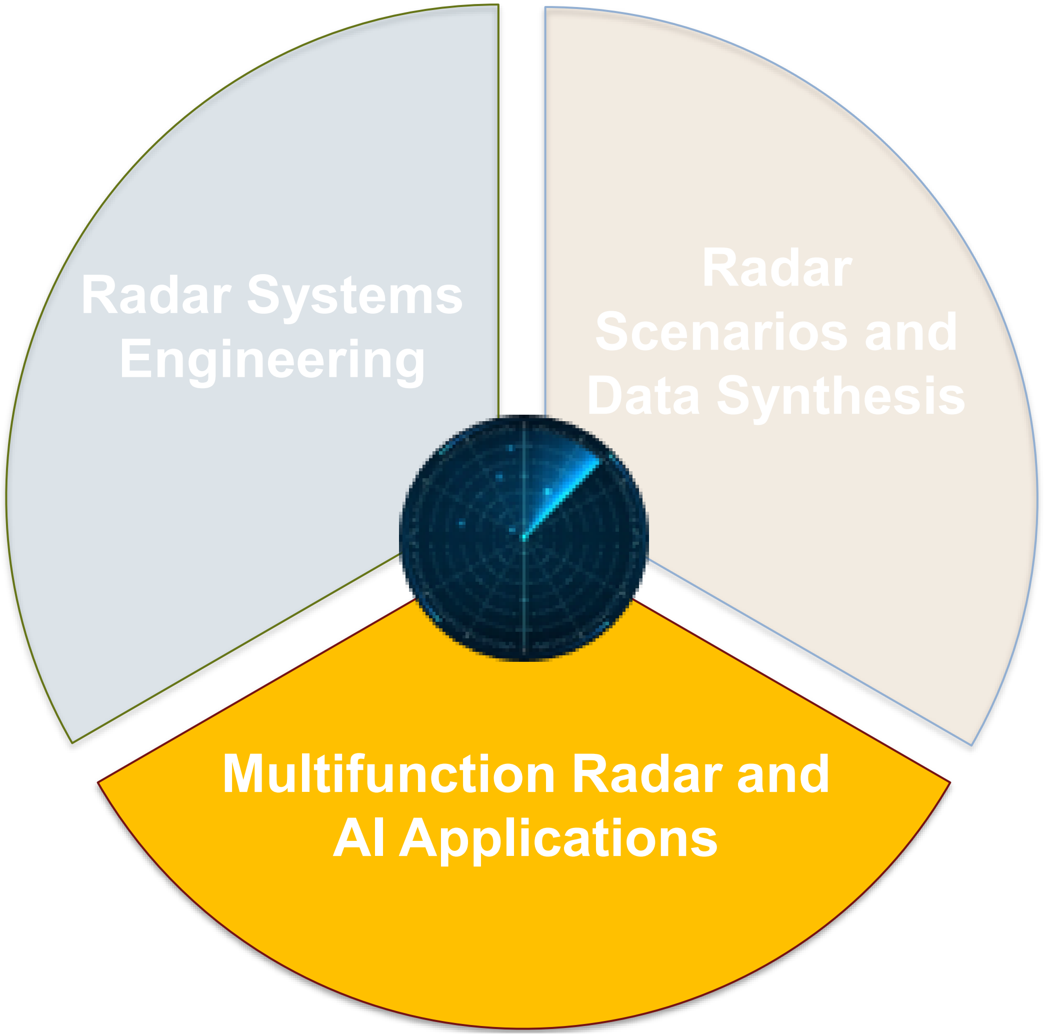
Form Bistatic Detections



- Perform signal processing
  - Range and Doppler processing
  - DP mitigation and beamforming
  - CFAR detection, clustering, and parameter estimation

- Simulates IQ for a passive bistatic scenario with 2 unknown transmitters
- Shows how to manually scan a bistatic transmitter

* *MBET – Monostatic-Bistatic Equivalence Theorem*


Radar Systems Engineering
Radar Scenarios and Data Synthesis
Multifunction Radar and AI Applications

Multifunction Radar and AI Applications

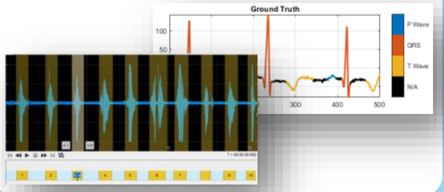
Cognitive Radar Workflow

Create and Access Datasets

Data sources

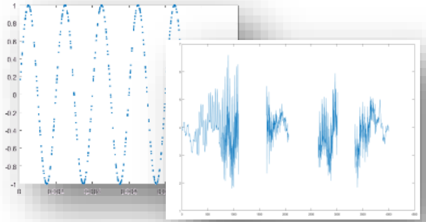
Simulation and augmentation

Data Labeling

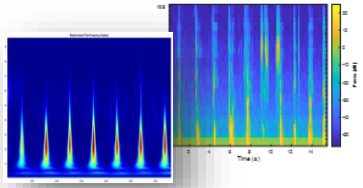


Preprocess and Transform Data

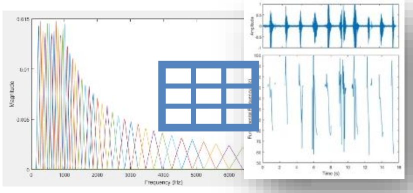
Pre-Processing



Transformation



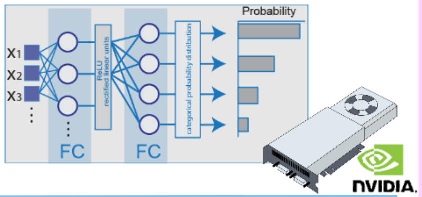
Feature extraction



Develop Predictive Models

Import Reference Models/ Design from scratch

Hardware-Accelerated Training



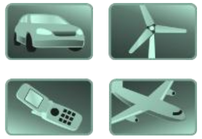
Analyze and tune hyperparameters

Accelerate and Deploy

Desktop Apps

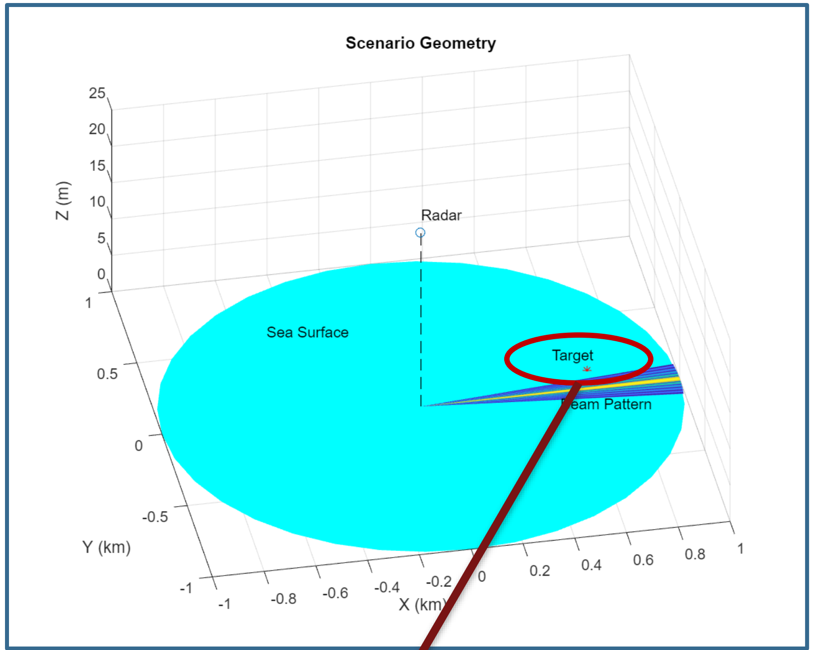
Enterprise Scale Systems

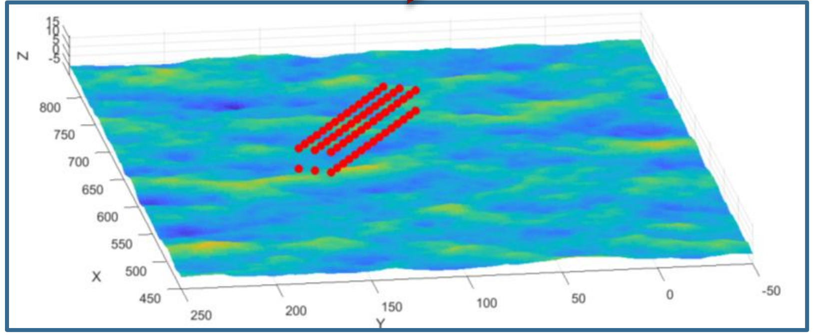
Embedded Devices and Hardware



Multifunction Radar and AI Applications

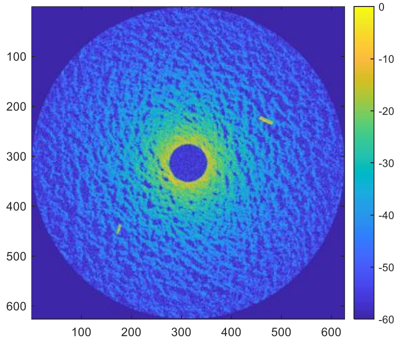
Remove Maritime Clutter with Neural Network



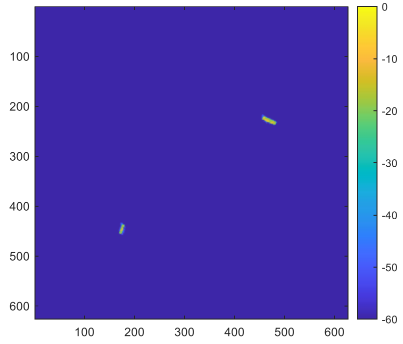


Training Data

Clutter + Target



Target Only

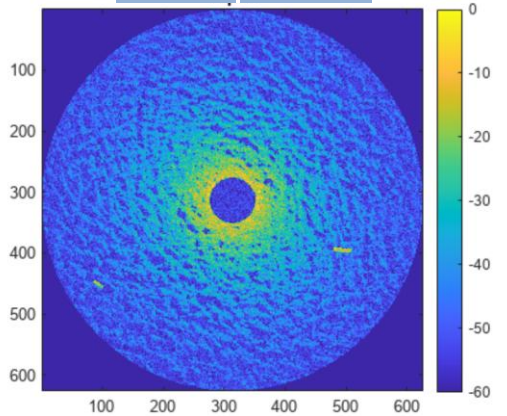


Variables in Training Data

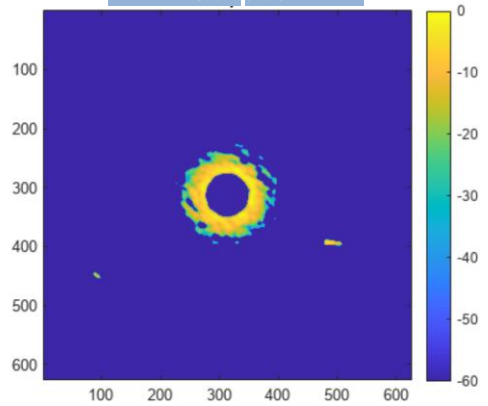
Surface		Target			
Wind Direction	Wind Speed	Position	Heading	Speed	RCS

Network Performance

Input



Output



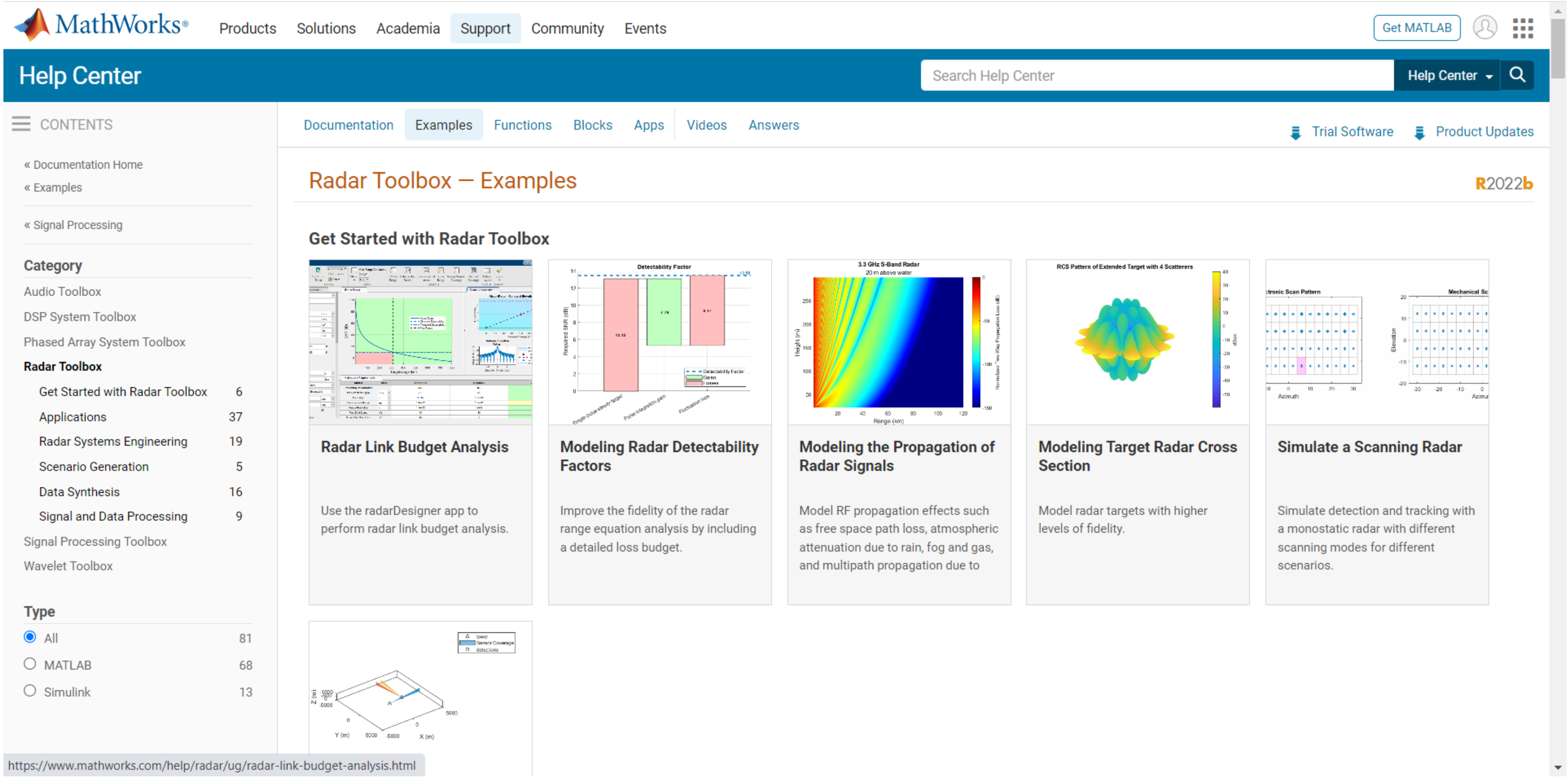
Waveform-level

[Link to Example]

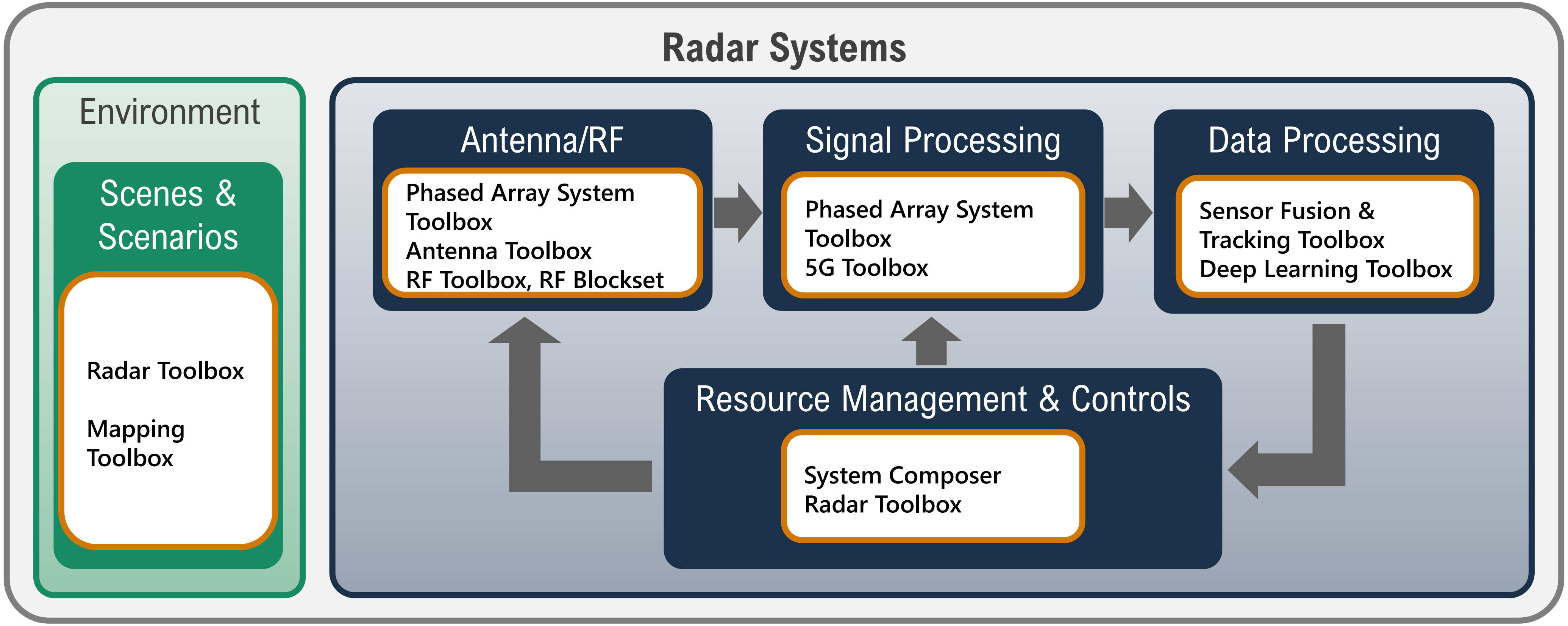
Summary

- **Perform link budget analysis and evaluate design trade-offs**
- **Simulate a radar scenario to synthesize radar data**
- **Example using AI workflow to remove maritime clutter**

Get Started with Radar Toolbox with Examples and Workflows



Develop Radar Systems with MATLAB and Simulink



Thank you!