

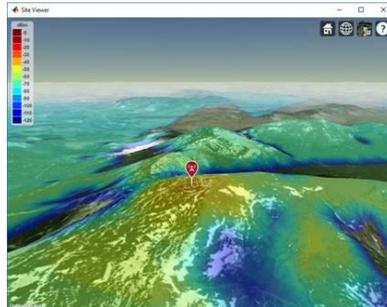
Introduction to Planar Antenna Design and Optimization

Giorgia Zucchelli – Product Manager RF & AMS

gzucchel@mathworks.com

Antenna Design, Analysis, and Visualization

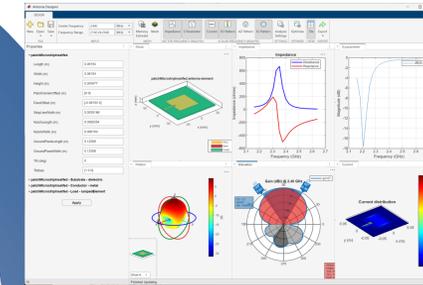
Propagation



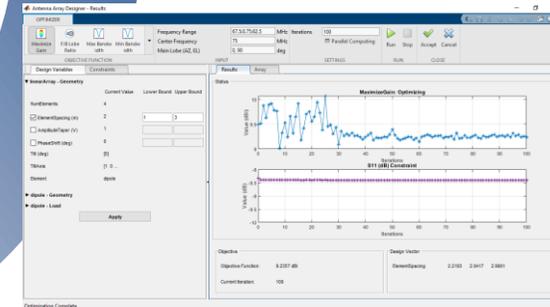
Design



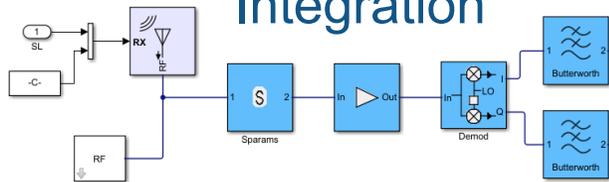
Analysis



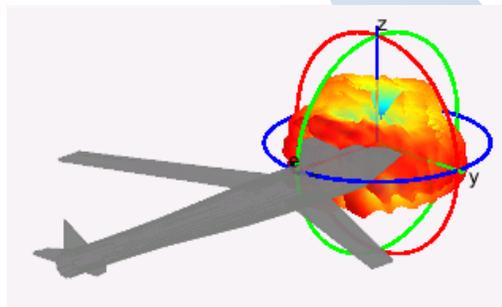
Optimization



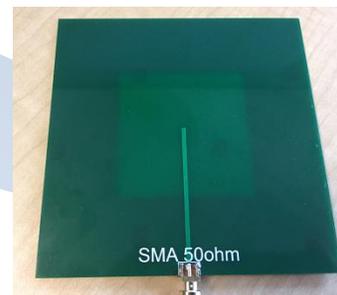
Integration



Installation



Fabrication



Antenna Design – Where To Start?



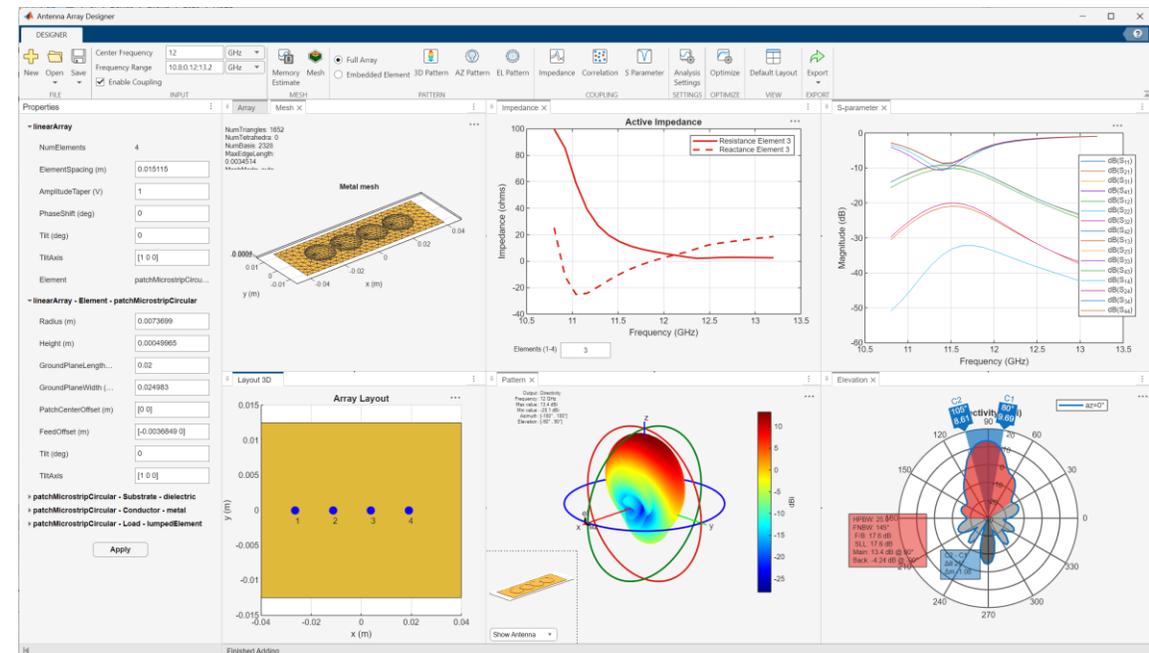
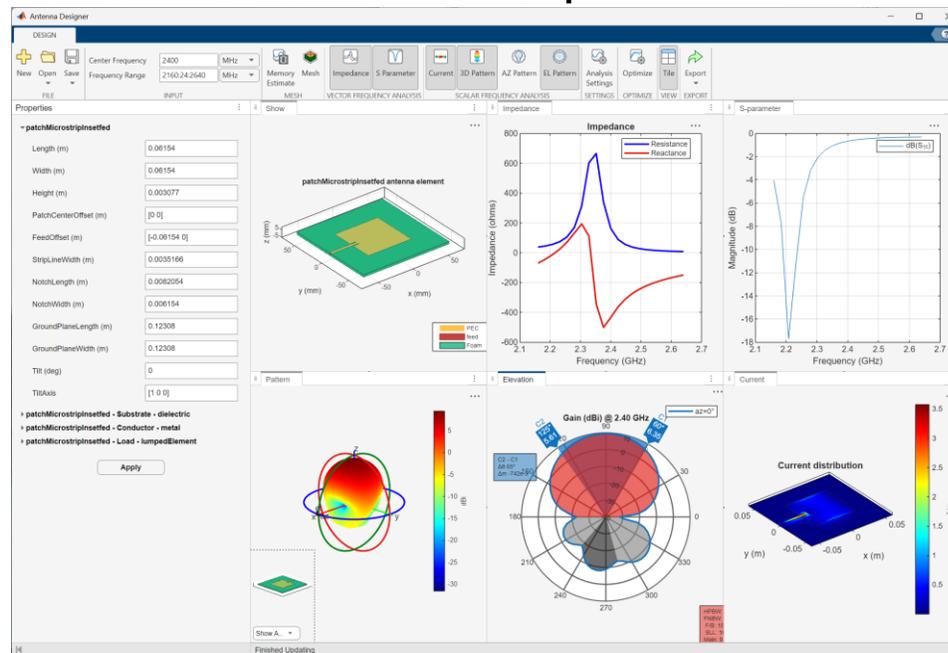
Antenna
Designer



Antenna Array
Designer

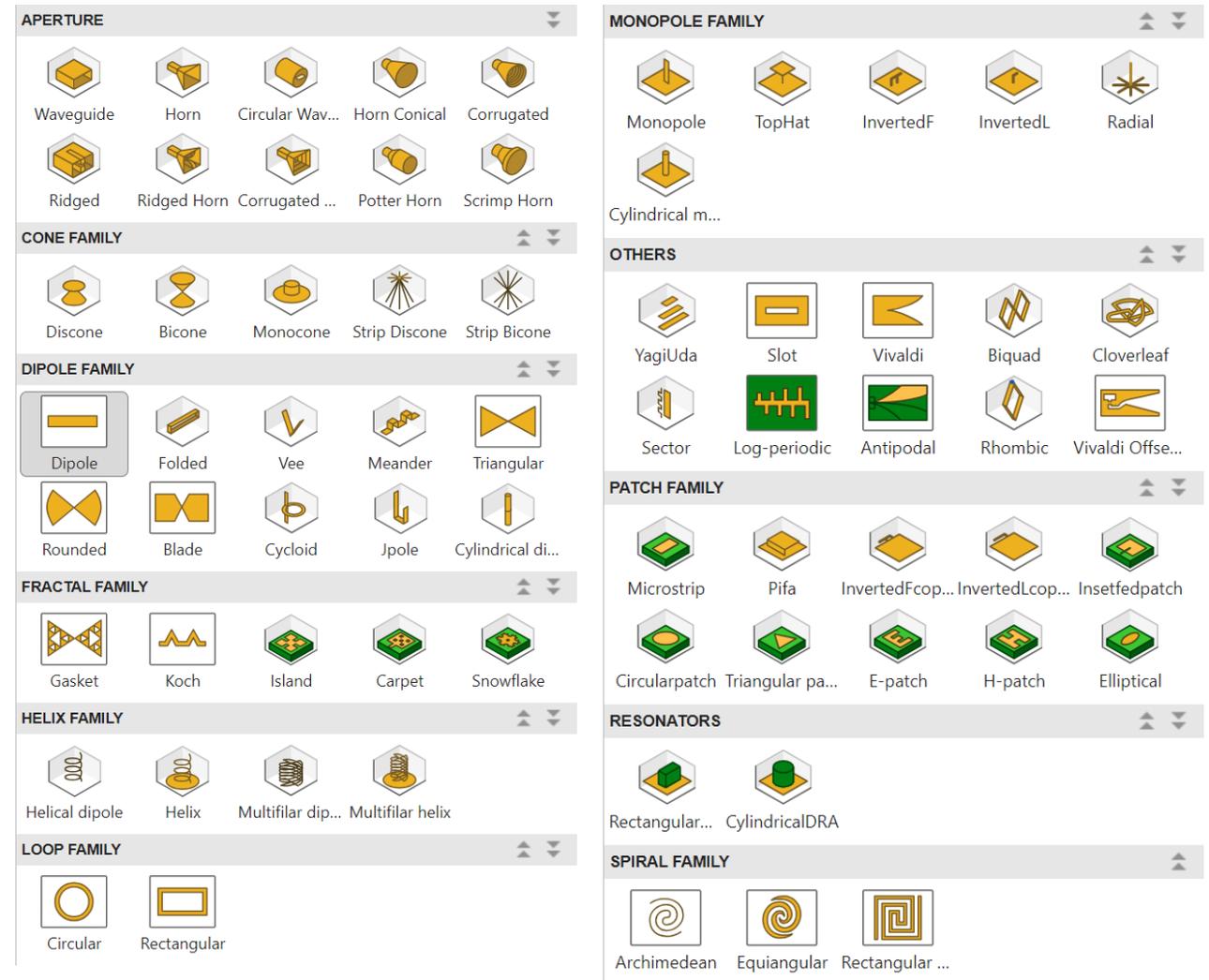
Antenna Designer / Array Designer

- Select an antenna / array based on the desired specifications
- Design the antenna / array at the operating frequency
- Visualize results and iterate on antenna geometrical properties
- Generate MATLAB scripts for automation



Antenna Catalog: Available Parameterized Geometries

- Dipole and bowtie antennas
- Monopole antennas
- Patch antennas
- Spiral, helix, and loop antennas
- Aperture and cone antennas
- Fractal antennas
- Dielectric resonators
- Other common antennas
- Backing structures

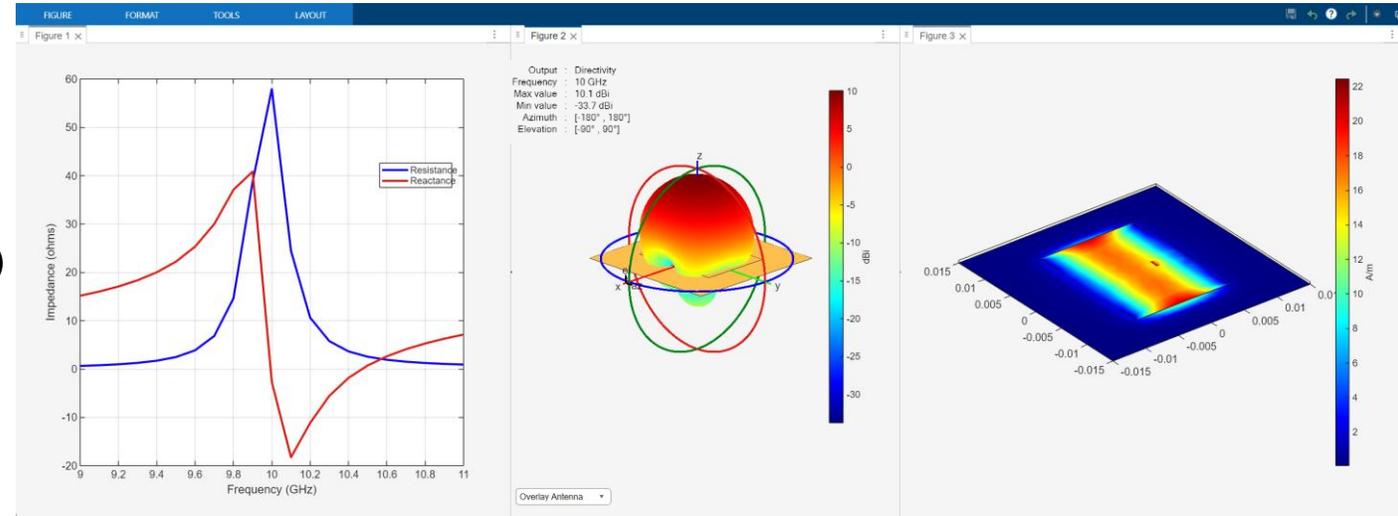
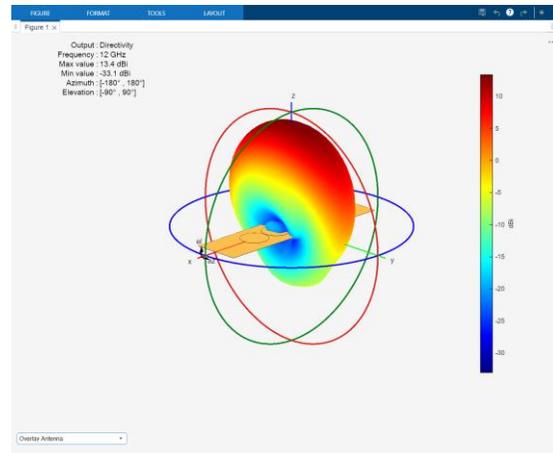


Building your First Antenna

```
p=design(patchMicrostrip,10e9)
impedance(p,linspace(9e9,11e9,21))
pattern(p,10e9)
current(p,10e9)
```

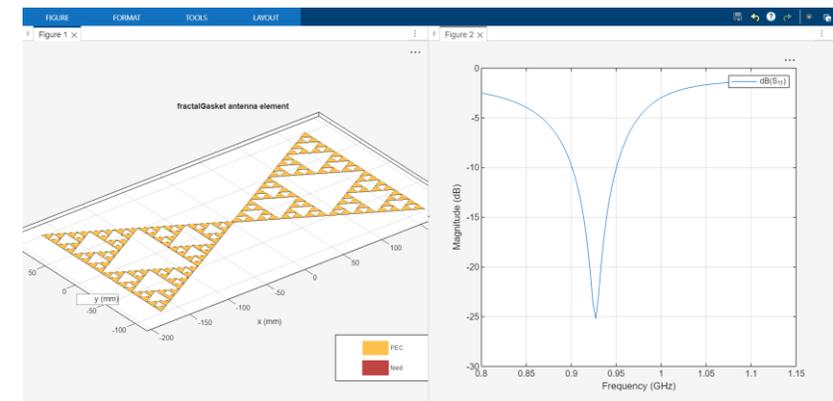
```
l=design(linearArray,12e9, ...
        patchMicrostripCircular)
```

```
l.NumElements=4
pattern(l,12e9)
```



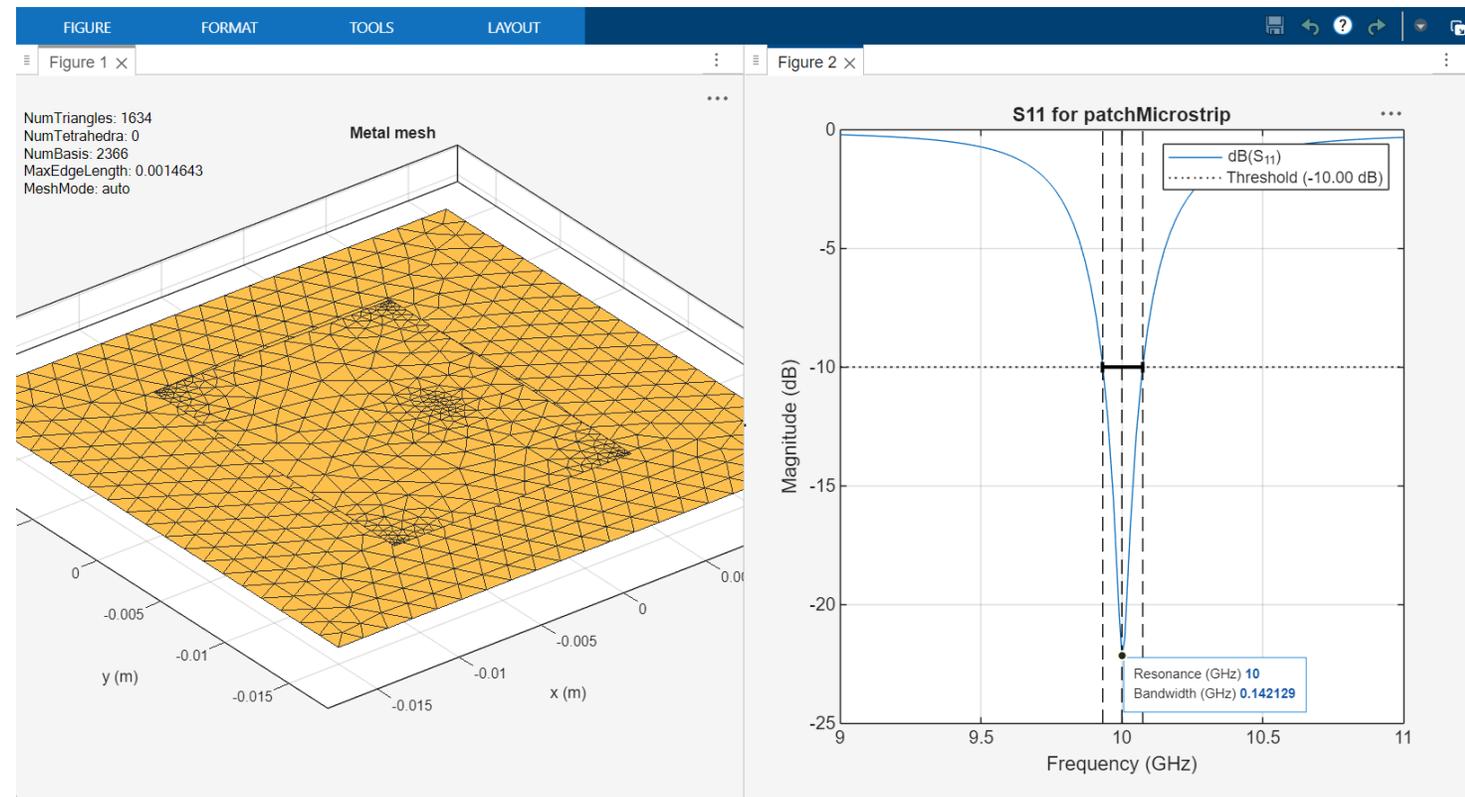
```
g=fractalGasket
g.NumIterations=4
s=sparameters(g,[800e6,1.1e9],...
              "SweepOption","interp")
```

```
rfplot(s)
```



Useful Tips and Tricks ...

- `p=design(patchMicrostrip,10e9)`
- `methods(p)`
- `s = sparameters(p, [9e9 11e9], "SweepOption", "interp")`
- `info(p)`
- `memoryEstimate(p)`
- `mesh(p)`

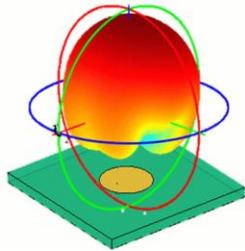


Enabling Antenna Design Automation

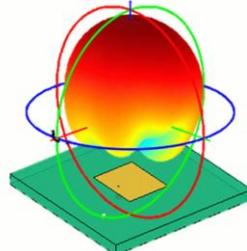
Benefits of a MATLAB-based catalog-driven approach:

- Rapidly evaluate different type of antennas
- Explore the design space programmatically
- Integrate with optimization / AI algorithms

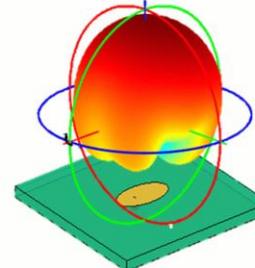
```
%% Circular Patch
r = 16e-3; % Radius
f2 = 9.25e-3; % Feed Offset
p_Circle= patchMicrostripCircular('Radius',r,'Height',h,...
'Substrate',d,'GroundPlaneLength',LGp,'GroundPlaneWidth',WGp,...
'FeedOffset',[-(r-f2) 0]);
pattern(p_Circle, Fc);
```



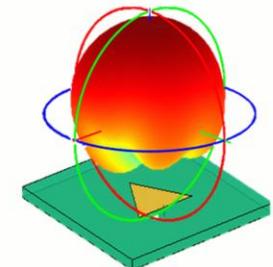
```
%% Rectangular Patch
l = 28.20e-3; % length
w = 34.06e-3; % width
f3 = 5.3e-3; % feedoffset
p_Rectangular= patchMicrostrip('Length',l,'Width',w,'Height',h,...
'Substrate',d,'GroundPlaneLength',LGp,'GroundPlaneWidth',WGp,...
'FeedOffset',[-(l/2-f3) 0]);
pattern(p_Rectangular, Fc);
```



```
%% Elliptical Patch
a = 33.5e-3; % Major Axis
b = 18.8e-3; % Minor Axis
f1 = 11.6e-3; % Feed Offset
p_Ellipse= patchMicrostripElliptical('MajorAxis',a,'MinorAxis',b,'Height',h,...
'Substrate',d,'GroundPlaneLength',LGp,'GroundPlaneWidth',WGp,...
'FeedOffset',[-(a/2-f1) 0]);
pattern(p_Ellipse, Fc);
```

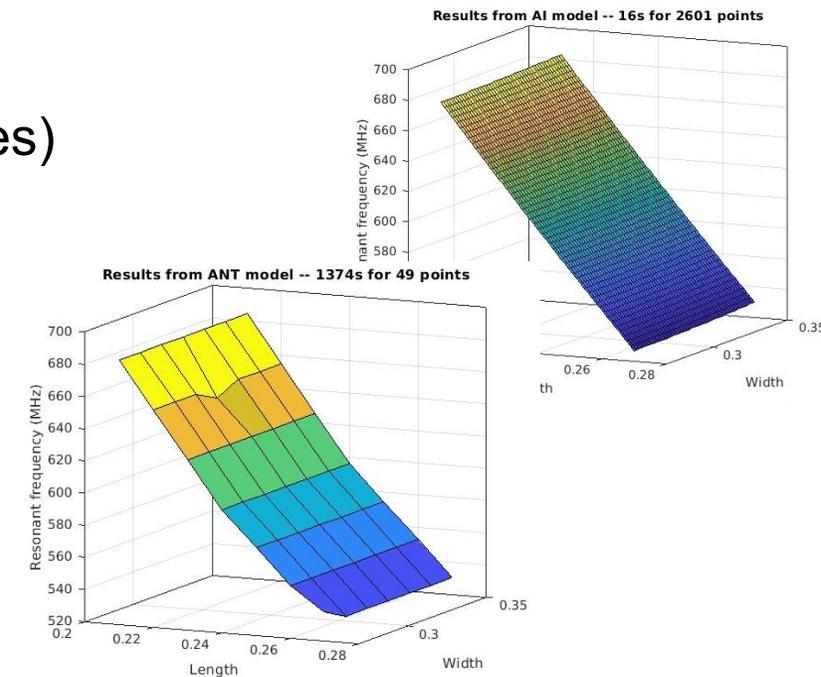
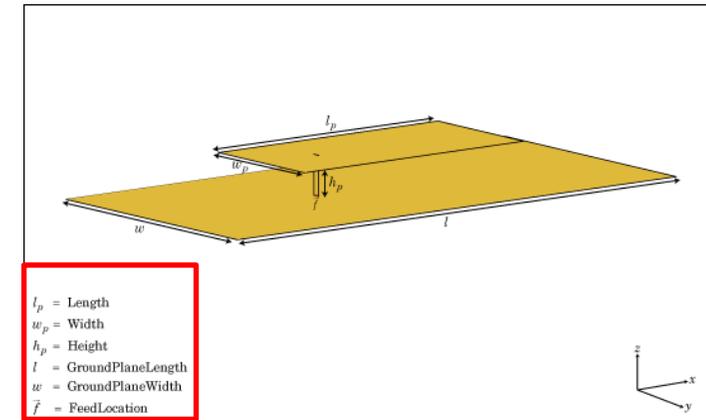


```
%% Triangular Patch
side = 37.63e-3;
f4 = 3.8e-3;
p_Triang= patchMicrostripTriangular('side',side,'Height',h,...
'Substrate',d,'GroundPlaneLength',LGp,'GroundPlaneWidth',WGp,...
'FeedOffset',[0 -side/2+f4]);
pattern(p_Triang, Fc);
```



Pretrained AI Antenna Models

- Unique solution
- Based on subset of antenna catalog
 - Ready-to-use models
 - Tunable parameters
 - Dipole, dipoleHelix, horn, waveguide, patch antennas (8 types)
- Enable AI-based analysis
 - Rapid characterization
 - Design space exploration
 - Resonant frequency, bandwidth, beamwidth, peak radiation

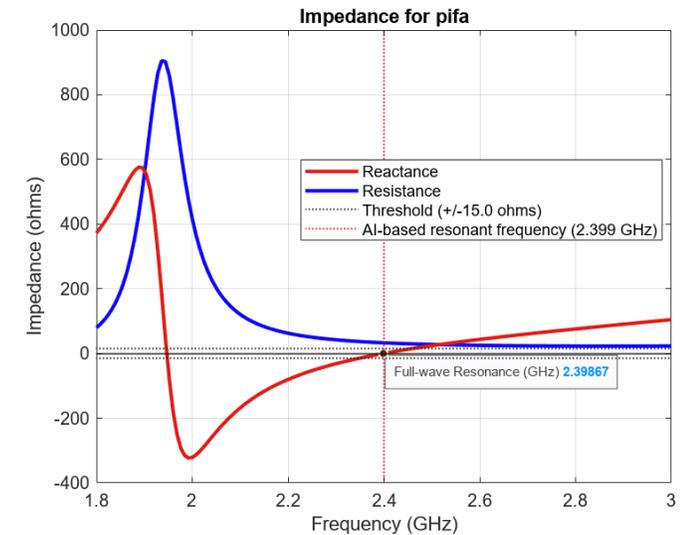
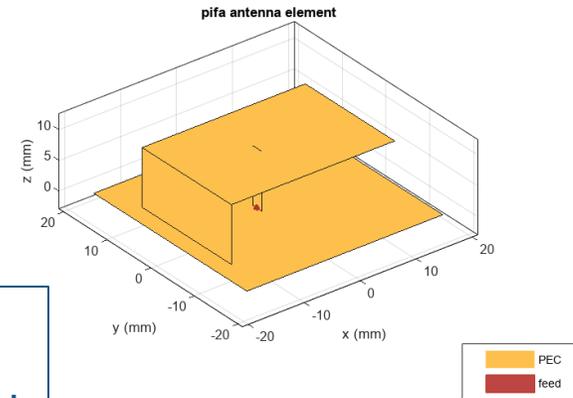


Rapid Design Exploration Using Pretrained AI Models

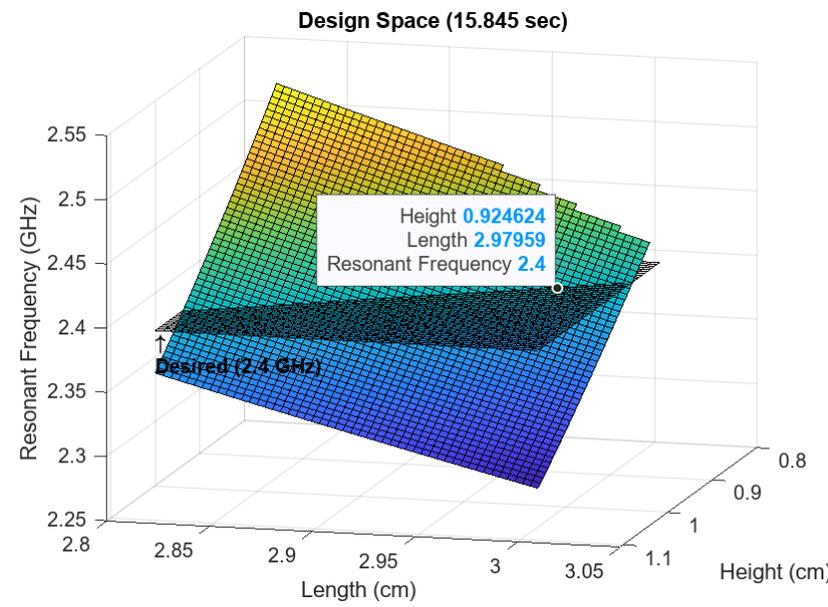
```
>> p = design(pifa, 2.4e9, ForAI=true)
p =
  AIAntenna with properties:

  Antenna Info
      AntennaType: 'pifa'
      InitialDesignFrequency: 2.4000e+09
  Tunable Parameters
      Length: 0.0292
      Width: 0.0194
      Height: 0.0097
      ShortPinWidth: 0.0194
  Show read-only properties
```

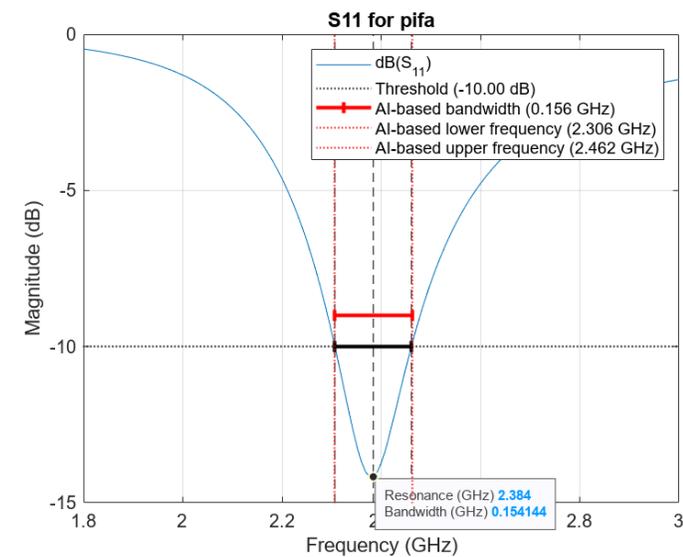
Iteratively repeat this after modifying tunable parameters!



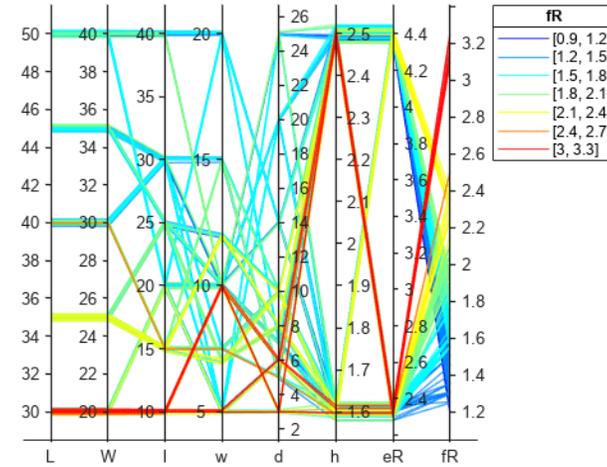
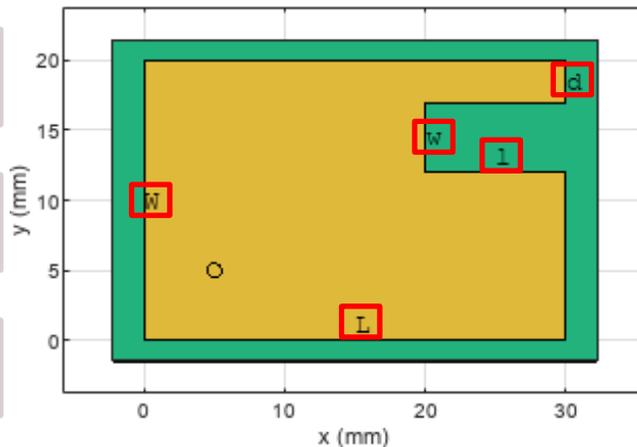
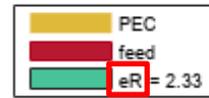
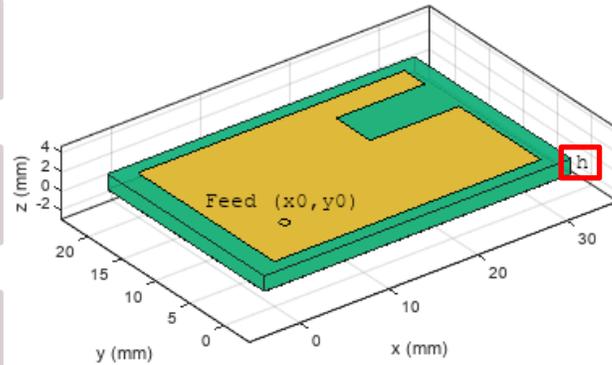
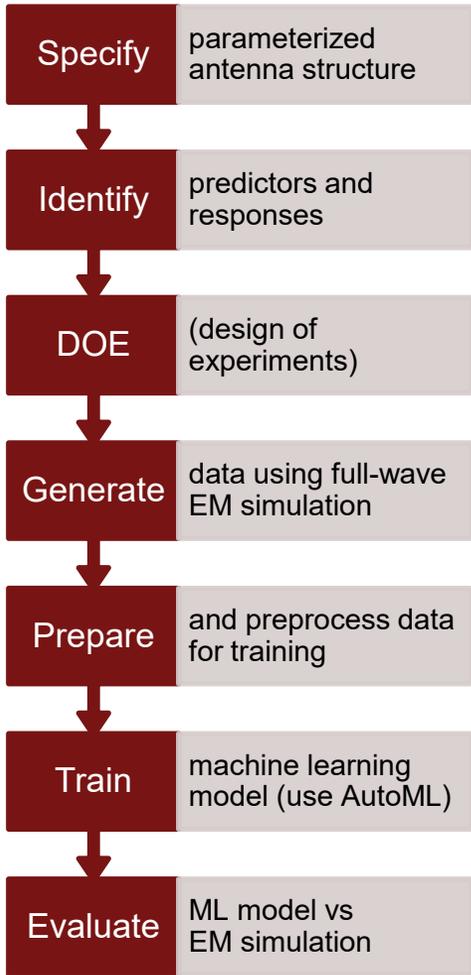
```
>> tic, fR = resonantFrequency(p), toc
fR =
  2.4091e+09
Elapsed time is 0.158043 seconds.
```



```
>> tic, [absBW, fL, fU, matching] = bandwidth(p), toc
absBW =
  1.4809e+08
fL =
  2.3186e+09
fU =
  2.4667e+09
matching =
  categorical
  Matched
Elapsed time is 0.190911 seconds.
```



Train your AI Antenna Model

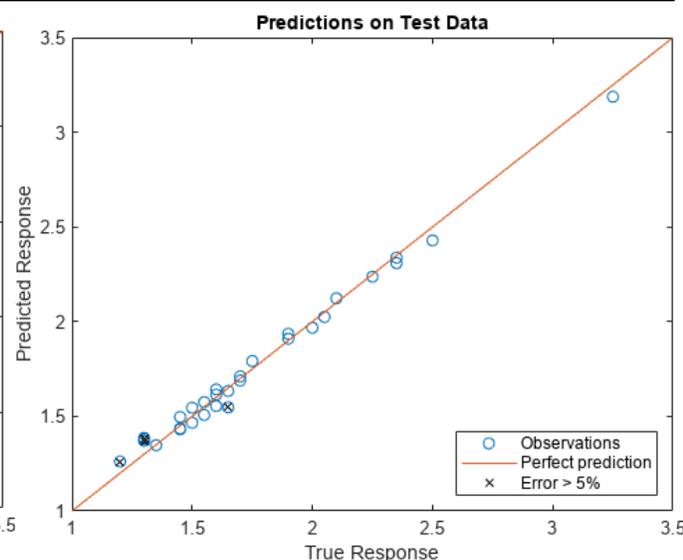
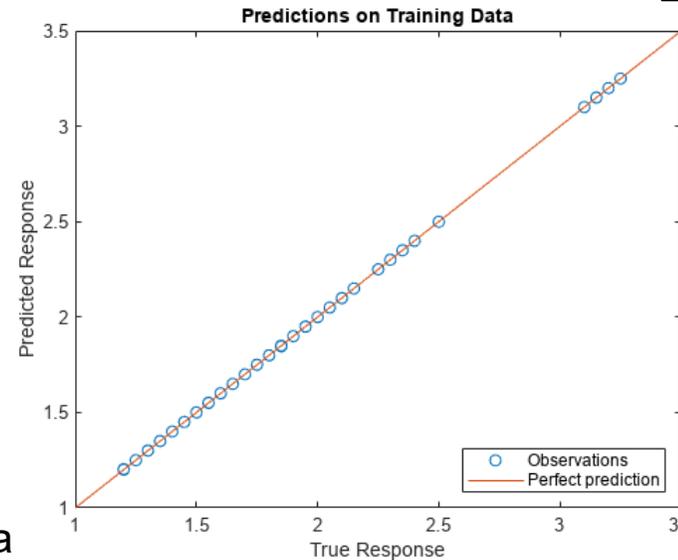


```

    Optimization completed.
    Total iterations: 60
    Total elapsed time: 370.6171 seconds
    Total time for training and validation: 252.4126 seconds

    Best observed learner is a gp model with:
    Learner: gp
    Sigma: 0.0073413
    BasisFunction: none
    KernelFunction: ardxponential
    KernelScale: NaN
    Standardize: false
    Observed log(1 + valloss): 0.023284
    Time for training and validation: 1.2777 seconds

    Best estimated learner (returned model) is a gp model with:
    Learner: gp
    Sigma: 0.0073413
    BasisFunction: none
    KernelFunction: ardxponential
    KernelScale: NaN
    Standardize: false
    Estimated log(1 + valloss): 0.023767
    Estimated time for training and validation: 1.3343 seconds
  
```



Train Machine Learning Model for Analysis of Custom Antenna

<https://www.mathworks.com/help/antenna/ug/train-mlmodel-for-analysis-of-custom-antenna.html>

Dielectric and Metal Material Properties

Dielectric materials:

- Use the dielectric catalogue of existing materials
- Define your own: relative permittivity, loss tangent

Dielectric	Relative permittivity	Loss Tangent
Air	1	0
Other	>1 (typically <10)	>0 (typically ~1e-3)

Metal materials:

- Perfect Electrical Conductor (PEC)
 - Zero thickness, infinite conductivity
- Specify the metal material from the catalog
- Define your own: thickness, conductivity

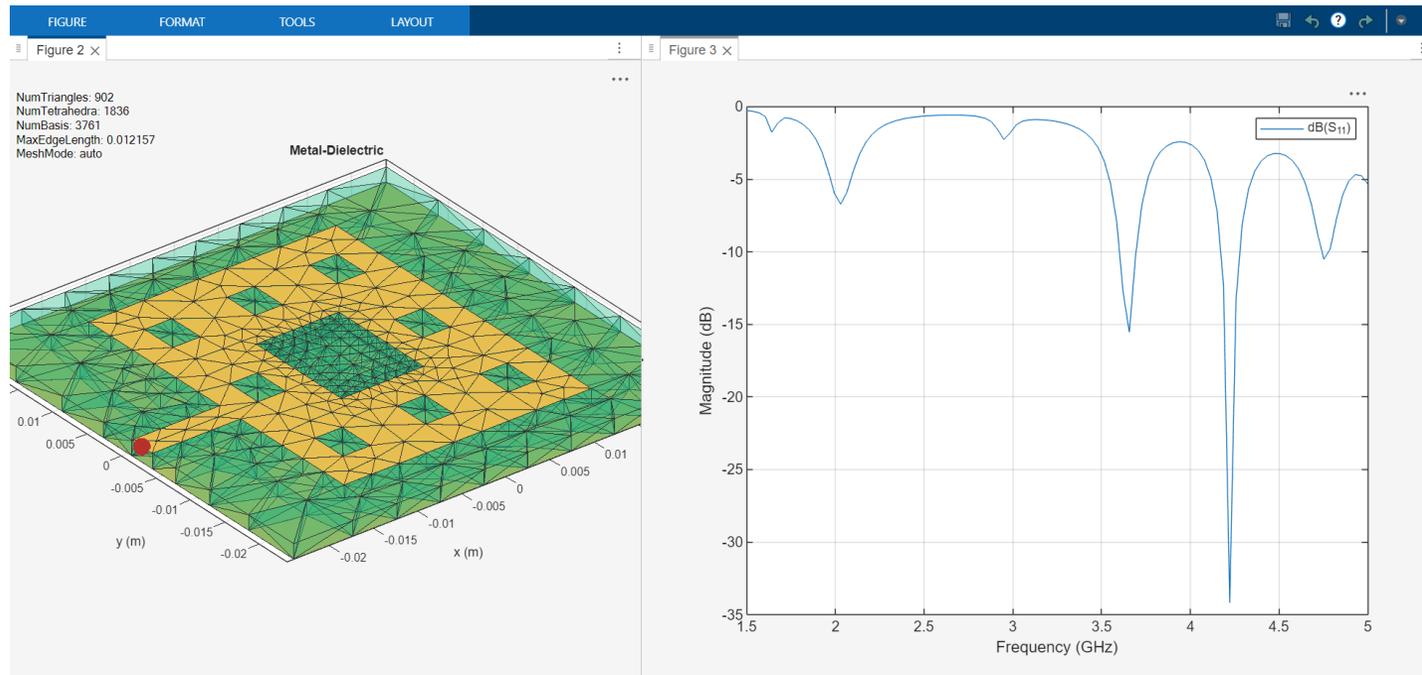
	Name	Relative_Permittivity	Loss_Tangent	Frequency	Comments
1	Air	1	0	1.0000e+09	
2	FR4	4.8000	0.0260	100000000	
3	Teflon	2.1000	2.0000e-04	100000000	
4	Foam	1.0300	1.5000e-04	50000000	
5	Polystyrene	2.5500	1.0000e-04	100000000	
6	Plexiglas	2.5900	0.0068	1.0000e+10	
7	Fused quartz	3.7800	1.0000e-04	1.0000e+10	
8	E glass	6.2200	0.0023	100000000	
9	RO4725JXR	2.5500	0.0022	2.5000e+09	
10	RO4730JXR	3	0.0023	2.5000e+09	
11	TMM3	3.4500	0.0020	1.0000e+10	
12	TMM4	4.7000	0.0020	1.0000e+10	
13	TMM6	6.3000	0.0023	1.0000e+10	
14	TMM10	9.8000	0.0022	1.0000e+10	
15	TMM10i	9.9000	0.0020	1.0000e+10	
16	Taconic RF-35	3.5000	0.0018	1.9000e+09	

	Name	Conductivity	Thickness	Units	Comments
1	PEC	Inf	0	m	
2	Copper	59600000	1.4000	mil	1 oz
3	Aluminium	37700000	30	mil	
4	Gold	41100000	0.2000	um	
5	Silver	63000000	0.2000	um	
6	Zinc	16900000	4	mil	
7	Tungsten	17900000	0.2000	um	
8	Lead	4550000	0.2000	um	
9	Iron	10000000	0.2000	um	
10	Steel	6990000	0.6800	mm	
11	Brass	15900000	0.6800		

PEC	Copper	Aluminium	Gold	Silver	Custom
• $\sigma = \text{Inf}$ • $d = 0$	• $\sigma = 59.6e06 \text{ S/m}$ • $d = 3.556e-05 \text{ m}$	• $\sigma = 37.7e06 \text{ S/m}$ • $d = 7.62e-04 \text{ m}$	• $\sigma = 41.1e06 \text{ S/m}$ • $d = 2e-07 \text{ m}$	• $\sigma = 63e06 \text{ S/m}$ • $d = 2e-07 \text{ m}$	• $\sigma = \text{User defined}$ • $d = \text{User defined}$
Zinc	Tungsten	Lead	Iron	Steel	Brass
• $\sigma = 16.9e06 \text{ S/m}$ • $d = 1.016e-04 \text{ m}$	• $\sigma = 17.9e06 \text{ S/m}$ • $d = 2e-07 \text{ m}$	• $\sigma = 4.55e06 \text{ S/m}$ • $d = 2e-07 \text{ m}$	• $\sigma = 10e06 \text{ S/m}$ • $d = 2e-07 \text{ m}$	• $\sigma = 6.99e06 \text{ S/m}$ • $d = 0.68e-03 \text{ m}$	• $\sigma = 15.9e06 \text{ S/m}$ • $d = 0.68e-03 \text{ m}$

Antenna Electromagnetic Analysis

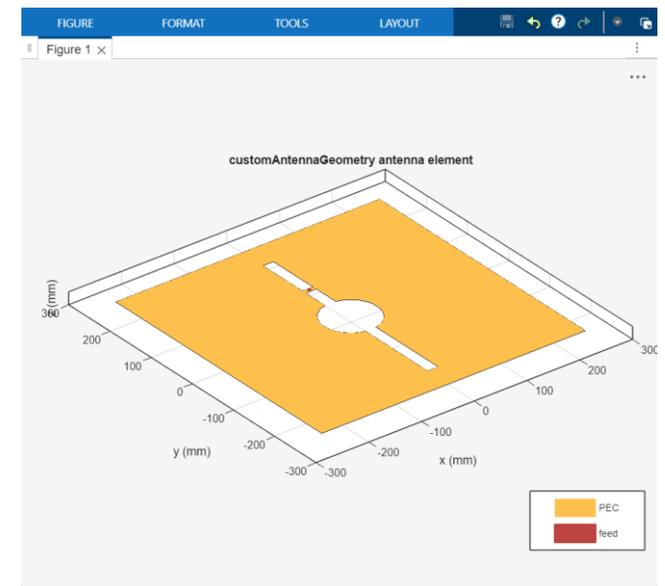
- Full-wave solver: Method of Moments
- Analysis-driven automatic mesh generation: triangles for surface, tetrahedra for volume
 - Manual control on meshing for advanced use cases
- Validated against state-of-the-art results



Design Custom Planar 2D Elements

- Define the boundary of your custom planar (2D) metal layer
 - Basic shapes: rectangle, circle, ellipse, polygon
 - Import boundary from photo
 - Operations: intersection, union, difference

```
plate = antenna.Rectangle('Length',0.16,'Width',0.16);  
notch1 = antenna.Circle('Center',[0,0],'Radius',.01);  
notch2 = antenna.Rectangle('Length',0.15,'Width',.005);  
b = plate-notch1-notch2;
```



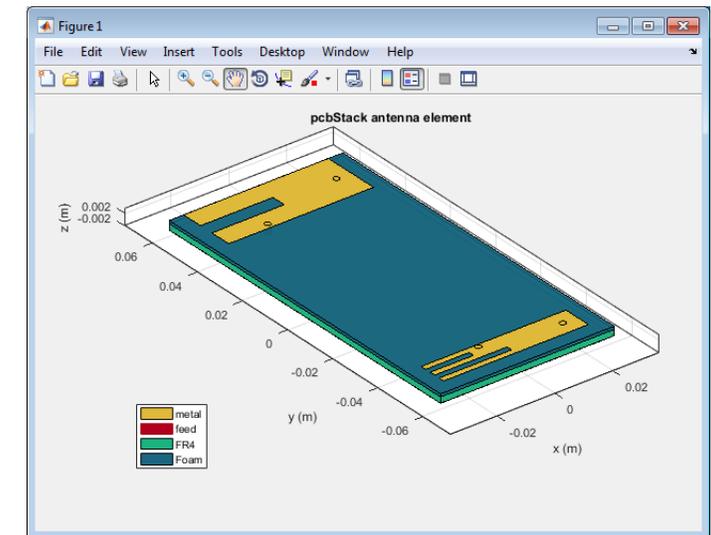
Design Custom Planar 2D Elements

- Define the boundary of your custom planar (2D) metal layer
 - Basic shapes: rectangle, circle, ellipse, polygon
 - Import boundary from photo
 - Operations: intersection, union, difference
- Add arbitrary dielectric and metal layers to PCB stack
 - Define vias and feed structures

```

p = pcbStack;
p.BoardShape      = b;
p.BoardThickness  = 3e-3;
p.Layers          = {ant, d1, d2, b};
p.FeedLocations   = [0.02 0.05 1 4; 0.02 -0.05 1 4];
p.FeedDiameter    = 1e-3;
p.ViaLocations    = [-6e-3 -0.046 1 4; -10e-3 0.042 1 4];
p.ViaDiameter     = 1e-3;
  
```

Metal and dielectric layers



PCB Antenna Designer App

Shapes

Boolean operations

Vias, Feed, Load

Export

PCB Stack

Variables

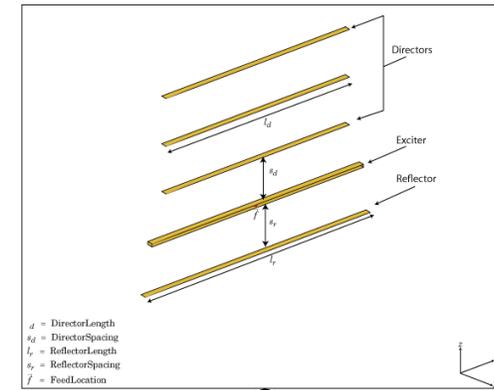
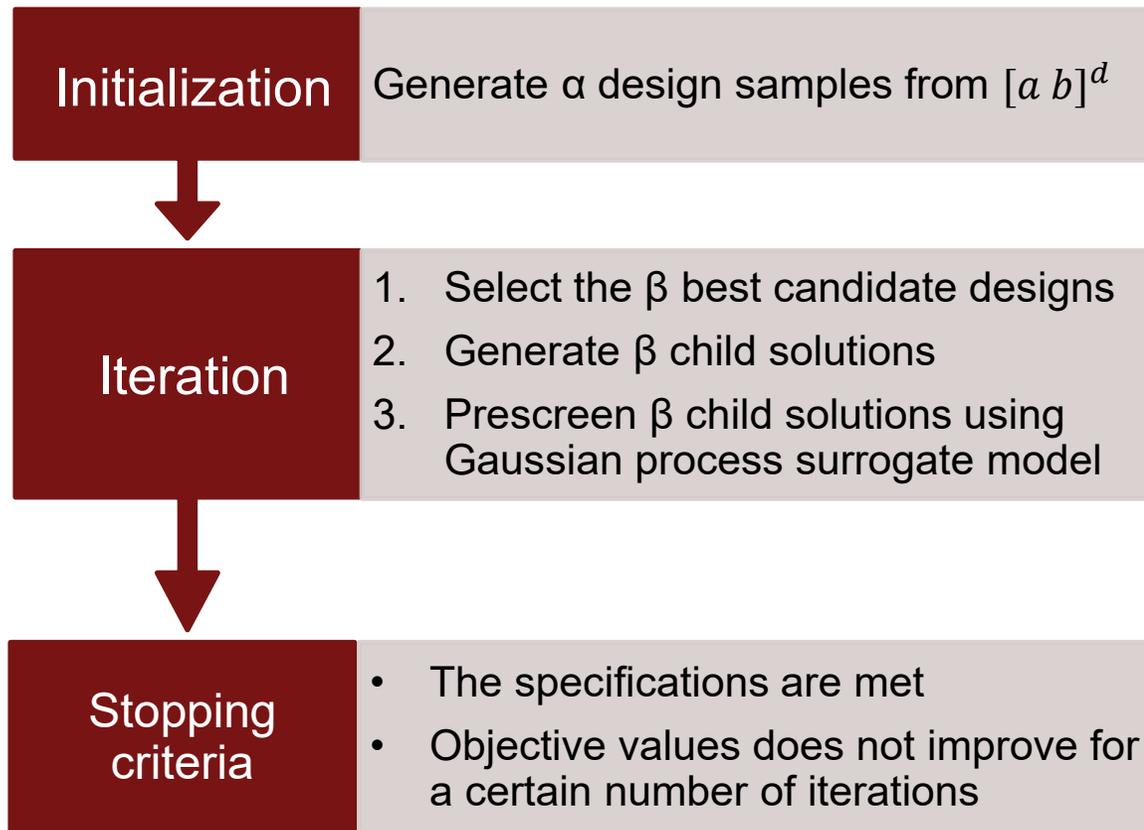
The screenshot displays the PCB Antenna Designer application interface. The top menu bar includes 'DESIGN' and 'ANALYSIS'. The 'DESIGN' menu contains options like 'New Session', 'Open Session', 'Save Session', 'Import', 'Add Layer', and 'Move Up/Down'. The 'ANALYSIS' menu includes 'Add Feed', 'Add Via', 'Add Load', 'Canvas Settings', 'Validate Design', 'Default Layout', and 'Export'. The 'SHAPES' toolbar shows 'Rectangle', 'Triangle', and 'Circle'. The 'FEED VIA' toolbar includes 'Add Feed', 'Add Via', and 'Add Load'. The 'ACTIONS' toolbar includes 'Intersect', 'Subtract', and 'Exclude'. The 'SETTINGS' toolbar includes 'Canvas Settings' and 'Validate Design'. The 'VIEW' toolbar includes 'Default Layout' and 'Export'. The 'PCB Stack' panel on the left shows a tree view of the design, including 'BoardShape', 'Layers', and 'Properties'. The 'Properties' panel for 'notch1' shows fields for Name, Angle, Center, Length, and Width. The 'Design Variables' table is also visible.

Name	Set Value/Expression	Derived Value
<input type="checkbox"/> L		2.1315 -
<input type="checkbox"/> W		2 -
<input type="checkbox"/> Lc	$9.3685 - W/2$	8.3685
<input type="checkbox"/> Wc	0.62663	-

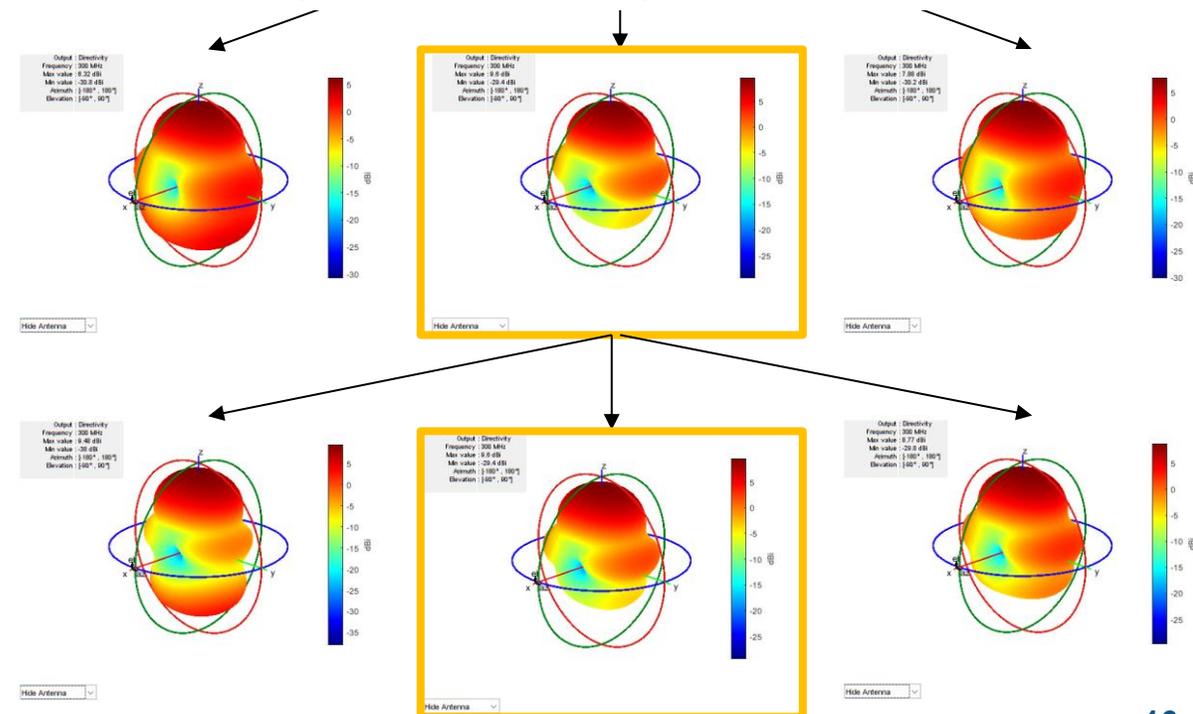
Surrogate-Based Optimization

Collaboration with Prof. Bo Liu, Univ. Glasgow

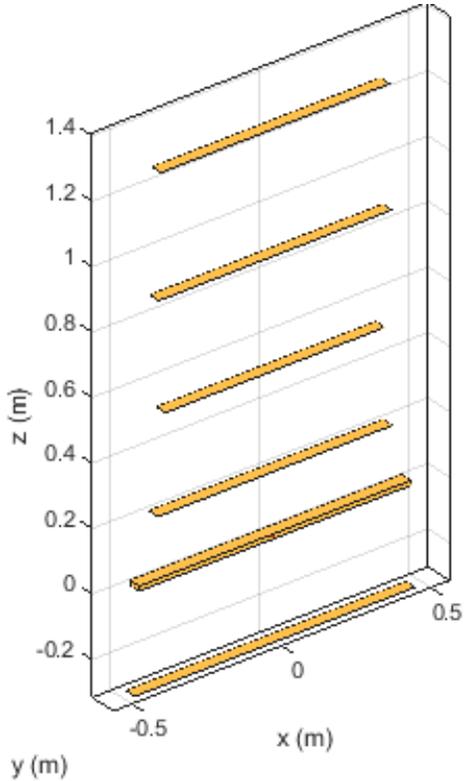
Surrogate model **A**ssisted **D**ifferential **E**volution for **A**ntenna synthesis (SADEA) is an AI-driven antenna design method



α design for reflector length from [0.1 1] meter

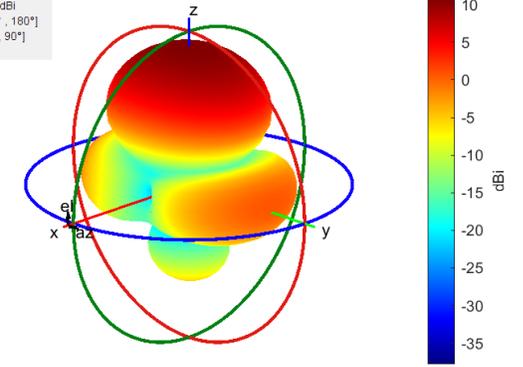


SADEA: Yagi-Uda Optimization Example

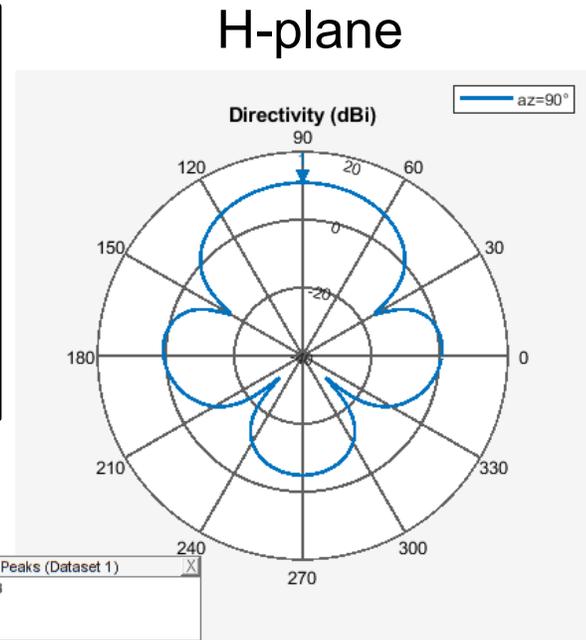
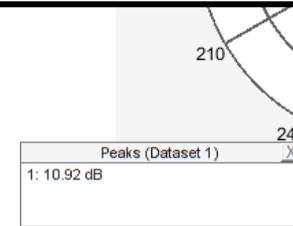


	initialdesignVal	optimdesignVal
1 Reflector Length	0.9085	0.9361
2 Director Length - 1	0.9085	0.7836
3 Director Length - 2	0.9085	0.7318
4 Director Length - 3	0.9085	0.7799
5 Director Length - 4	0.9085	0.7638
6 Reflector Spacing	0.5451	0.3166
7 Director Spacing - 1	0.4542	0.2054
8 Director Spacing - 2	0.4542	0.3085
9 Director Spacing - 3	0.4542	0.3501
10 Director Spacing - 4	0.4542	0.3888

Output : Directivity
 Frequency : 165 MHz
 Max value : 10.9 dBi
 Min value : -37.8 dBi
 Azimuth : [-180°, 180°]
 Elevation : [-90°, 90°]



	datasheetvals	optimdesignvals
Gain (dBi)	10.5	10.917
F/B	16	15.762
E-plane Beamwidth (deg.)	54	50
H-plane Beamwidth (deg.)	63	60
Impedance Bandwidth (%)	10	10



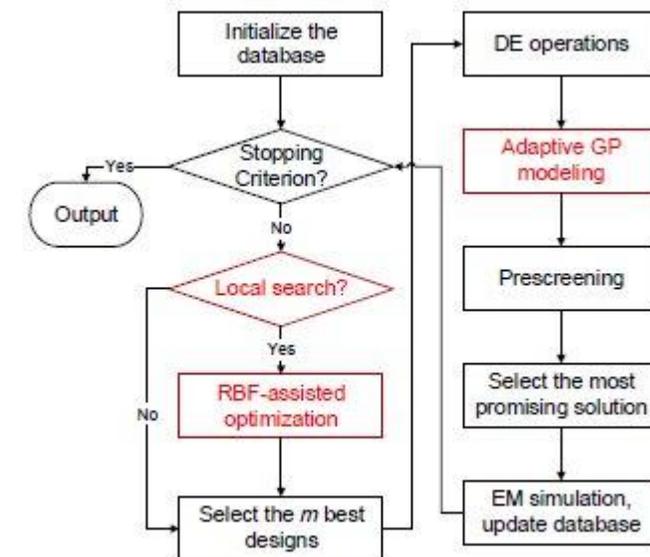
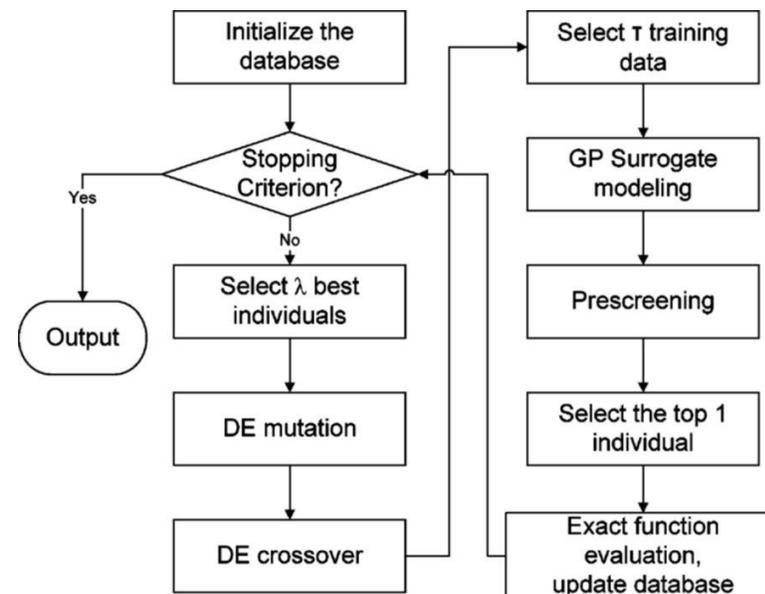
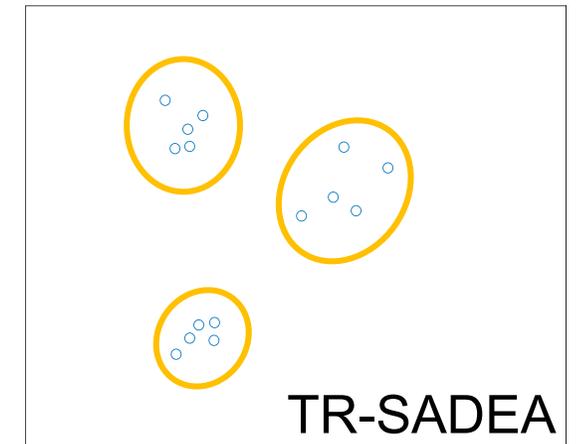
AI for Surrogate Based Optimization

TR (Training-cost Reduced) – SADEA

- Faster, while converging to a good solution
- Adaptive Gaussian Process predictions
- Handle up to 80 design variables
- Enabled in apps and command line



- Sample to be predicted
- GP model



Antenna Optimization - Summary



Antenna
Designer



Antenna Array
Designer



PCB Antenna
Designer

- Standard objective functions available in the apps
 - SADEA, TR-SADEA, Surrogate optimization
 - Support for geometric constraints and integration with RF PCB Toolbox
- Custom objective functions can be defined at the command line
 - Interface for SADEA and TR-SADEA available
 - Can be used with 3rd party EM tools (*)

The screenshot shows the 'Antenna Array Designer - Results' dialog box with the following settings:

- Frequency Range: 67.5:0.75:82.5 MHz
- Frequency: 75 MHz
- Iterations: 100
- Center Frequency: 75 MHz
- Main Lobe (AZ, EL): 0, 90 deg
- Objective Function: Maximize Gain

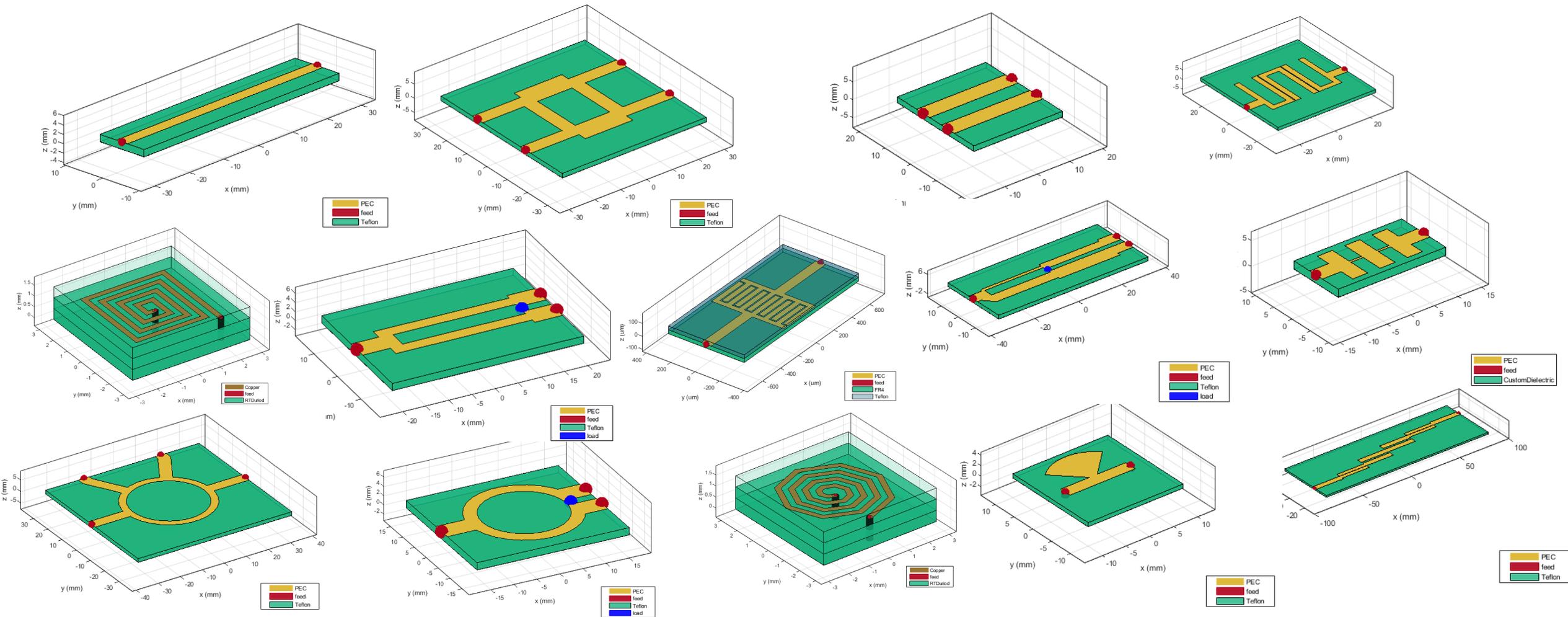
The objective function selection window shows the following options:

- MAXIMIZE:** Maximize Gain, F/B Lobe Ratio, Max Bandwidth
- MINIMIZE:** Min Bandwidth, Minimize SLL, Array Thinning, Minimize Area

```

patternoptions = psoptimset(@patternsearch);
patternoptions.PlotFcns = @psplotbestf;
patternoptions.MaxIter = 25;
optimdesign = patternsearch(@(x) yagi_objective_function(yagidesign,x,freq,elang),...
    parasitic_values, [], [], [], [], LB, UB, [], patternoptions);
  
```

Integrating with RF PCB Components



Example of RF PCB Design and Analysis: Branchline Coupler

```
c = couplerBranchline;
methods(c);
```

```
c = design(couplerBranchline, 2.4e9);
show(c);
```

```
s = sparameters(c, [1e9 3e9], ...
    "SweepOption", "interp")
rfplot(c);
```

```
>> c = design(couplerBranchline, 2.4e9)
```

```
c =
```

couplerBranchline with properties:

PortLineLength: 0.0116

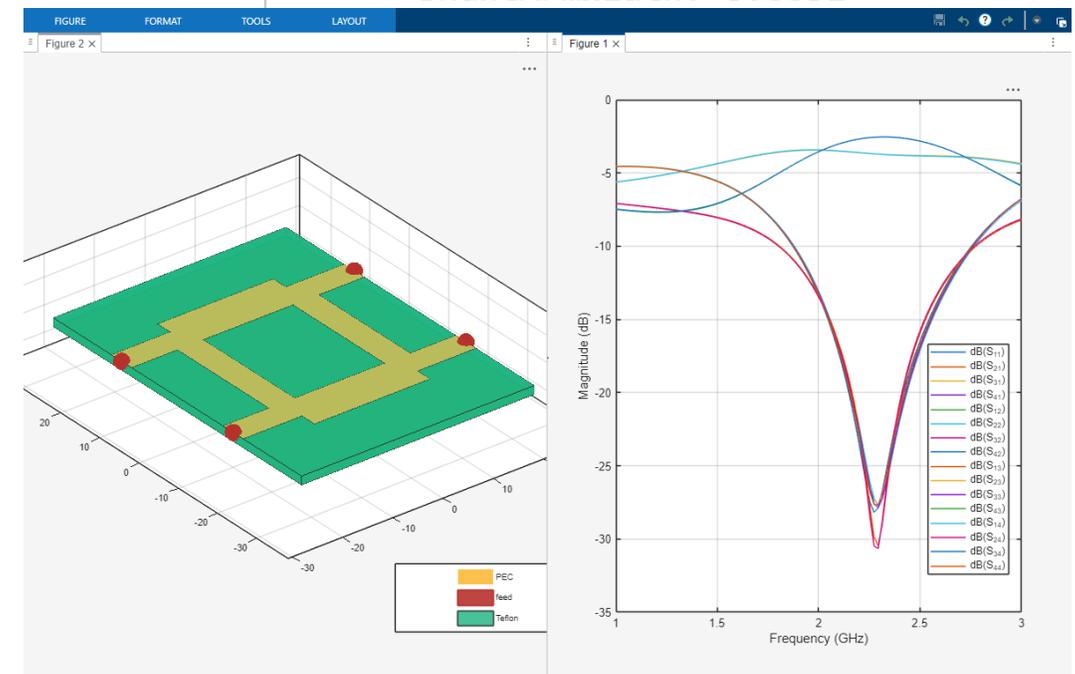
PortLineWidth: 0.0051

SeriesArmLength: 0.0229

SeriesArmWidth: 0.0083

ShuntArmLength: 0.0233

ShuntArmWidth: 0.0051

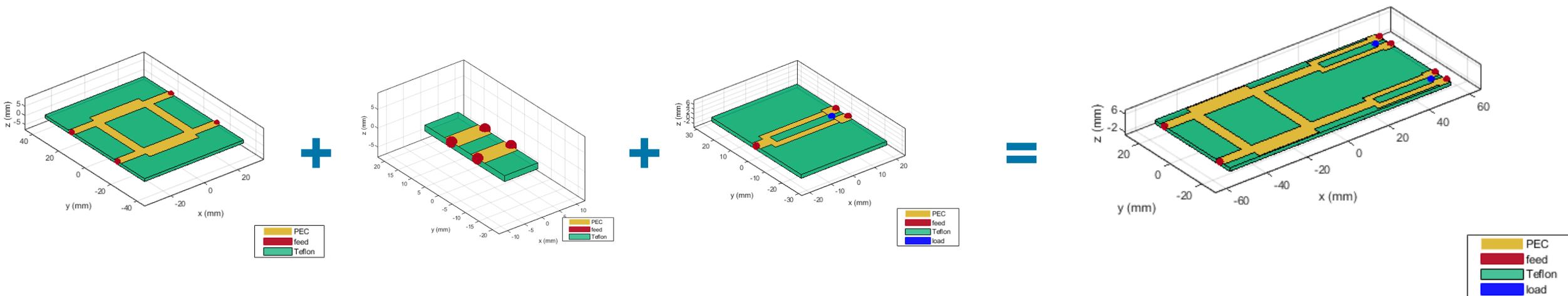


Cascading PCB Components

- Modular approach to building PCB assemblies
- Automatic port assignments
- Combine components (shapes) and antennas

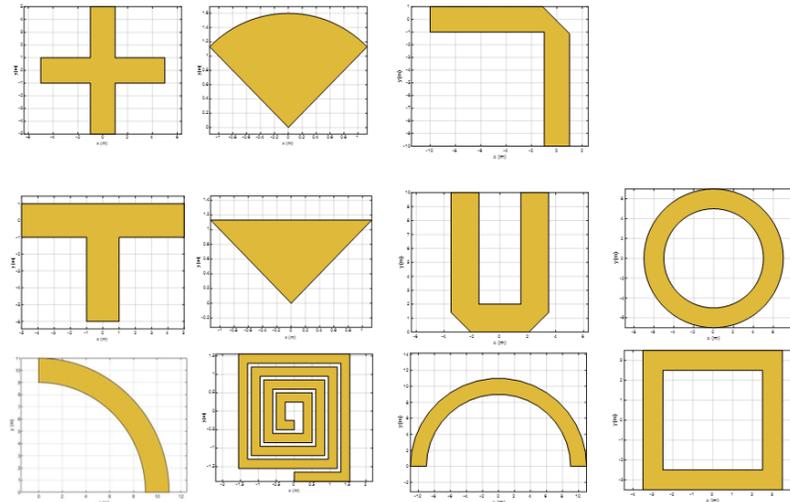
```
c = design(couplerBranchline,2.4e9);
mc = design(coupledMicrostripLine,2.4e9);
mc.Spacing = c.ShuntArmLength;
w = design(wilkinsonSplitter,2.4e9);
```

```
r = pcbcascade(c,mc);
r = pcbcascade(r,w,3,1);
r = pcbcascade(r,w,3,1);
show(r)
```



PCB Design Using Basic Shapes

- Design new components based on arbitrary shapes
 - Modify the geometry of existing components
1. Define the geometry of the basic shape
 2. Use variables for parametric analysis and facilitate design exploration
 3. Perform Boolean operations (add, subtract, intersect)
 4. Add layer to a PCB components



Bends

bendCurved
bendMitered
bendRightAngle

U Bends

ubendCurved
ubendMitered
ubendRightAngle

Rings, Curves, and Radial

ringAnnular
ringSquare
curve
radial
delta

Traces

traceSpiral
traceRectangular
traceTee
traceCross

Custom Traces

tracePoint
traceLine

Fabrication with Gerber File Generation

Design your PCB antenna / antenna array

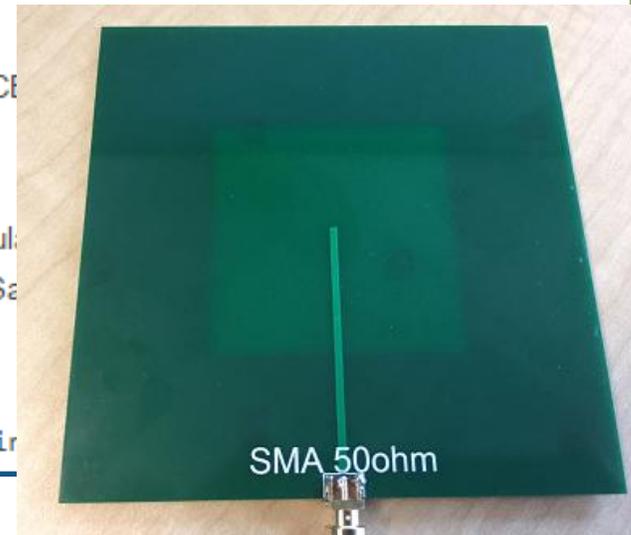
- Chose the manufacturing service
- Chose the connector
- Generate Gerber files

Type of service from PCB services package, specified as one of the following:

- AdvancedCircuitsWriter – Configure Gerber file generat
- CircuitPeopleWriter – -Configure Gerber file generation
- DirtyPCBsWriter – Configure Gerber file generation for
- EuroCircuitsWriter – Configure Gerber file generation fo
- GerberLookWriter – Configure Gerber file generation fo
- GerberViewerWriter – Configure Gerber file generation f

Type of connector from PCB connector package, specified as one of the following:

- Coax Connectors - Coax RG11, RG174, RG58, and RG59 connectors directly soldered to PCB
- IPX Connectors - LightHorse IPX SMT jack or plug surface mount RF connector.
- MMCX Connectors - MMCX Cinch or Samtec surface mount RF connectors.
- SMA Connectors - Generic 5-pad SMA surface mount RF connectors, with 4 corner rectangular
- SMAEdge Connectors- Generic SMA edge-launch surface mount RF connector. Cinch and Sa
- SMB Connectors - Johnson/Emerson and Pasterneck SMB surface mount RF connectors.
- SMC Connectors - Pasterneck SMC and SMC edge-launch surface mount RF connectors.
- Coaxial Cable Connectors - Semi-rigid 0.020 inch, 0.034 inch, 0.047 inch, and 0.118 in



Summary: Design Planar Antennas and RF PCBs

- Use a catalog of components (antennas and PCC elements)
- Build and optimize your own planar geometry
- Perform EM analysis and speed up with AI
- Integrate with feed and matching networks
- Generate Gerber files for fabrication

