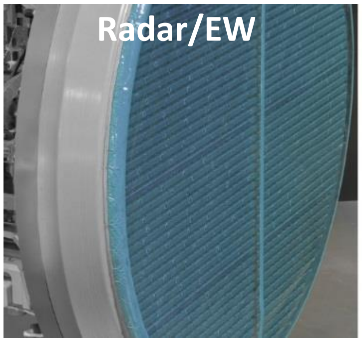
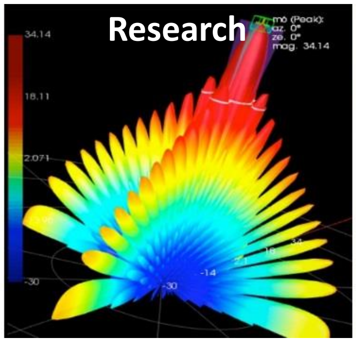




PHASED ARRAY SYSTEM DESIGN & SIMULATION USING KEYSIGHT SYSTEMVUE 2017

Abstract

The Phased Array Beamforming Kit in SystemVue provides system architects in 5G, Radar/EW and Satellite communications with the essential tools to evaluate phased array and beamforming subsystems, including RF, Digital, and Hybrid beamforming architectures. This workshop intends to provide a hands-on learning experience on Digital Beamforming, RF Phased Array analysis and Antenna Analysis in order to design Phased Array Systems (PAS) keeping many practical aspects of PAS into consideration that designers can use SystemVue platform for their Phased Array System designs with higher degree of confidence.

Anurag Bhargava & Pratik Khurana, Keysight Technologies, India

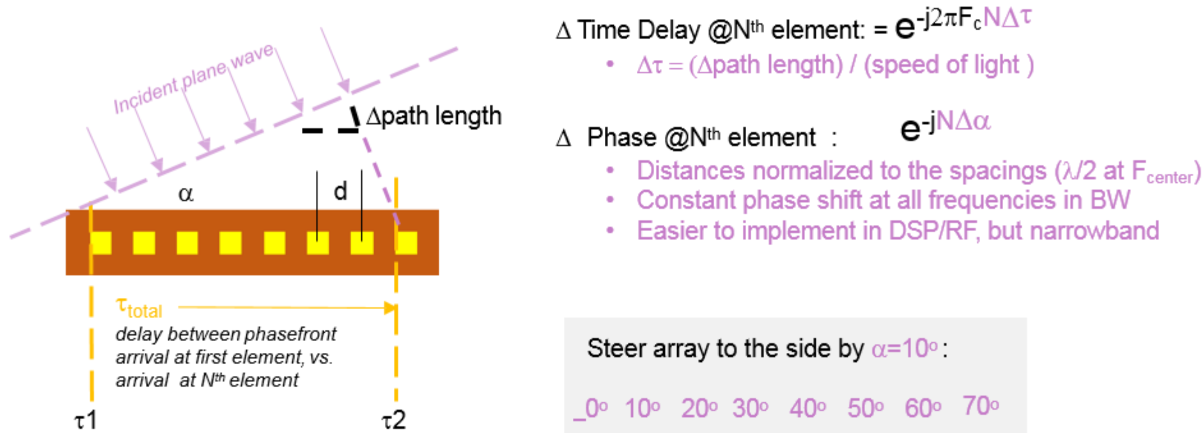
Introduction

Phased array has been used in military for several decades, why are we talking about it in this workshop? because we see a transitioning now to the commercial world, since these platforms have becoming low enough cost, and the consumer demand is pulling the need, I mean our voracious appetite for data and services from our cell phones or any wireless devices. The cost, the demand, and the trend to go for millimeter wave making it worthwhile to investigate and invest in phased array.

By using phased array, you are focusing energy on a specific direction in space, which reduces interference and increase your range, especially in mmwave channels. Using phased array, you are able to not radiate energy into unwanted direction, and you improve security and lowering interference susceptibility for yourself.

In addition, a big enough array can provide multiple concurrent services, in military, these can be multi-functional communication as well as electronic warfare, and in fact, it exists already for several years. For 5G mobility, you are looking at a very sophisticated MIMO and BF techniques to provide services for specific handsets at different directions simultaneously reusing the same frequency band. So, this is all great, however it takes a lot of effort to design these phased arrays with multiple disciplines involved.

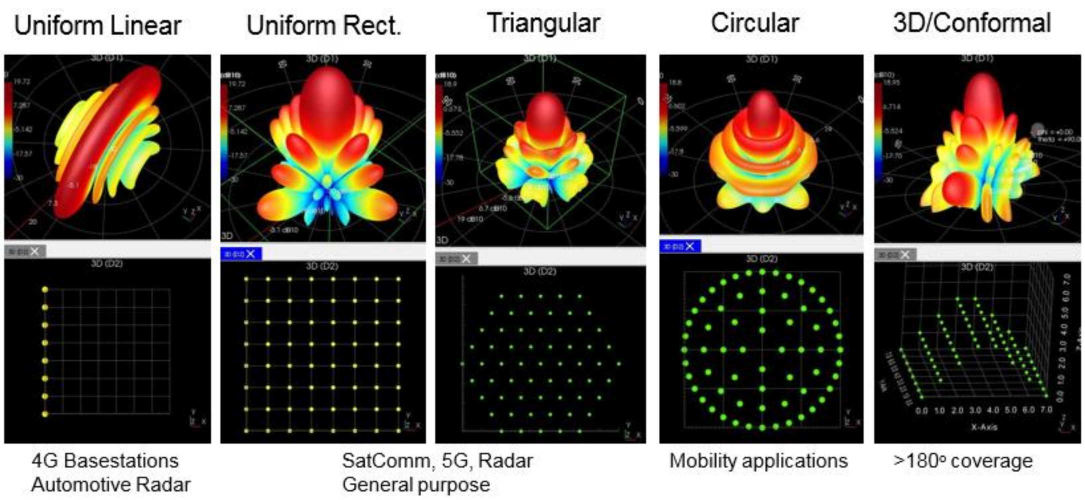
In Phased Array Radar system, the beam is steered by changing the phase of each antenna element for both transmission and receiving. We are recreating the plane wave either by using Phased Shifters or Delay Line.



Keysight SystemVue allows system designers to consider RF nonlinear and noise effects, Gain/Phase quantization, and Monte Carlo variations effects on total beam quality, sidelobe levels, and effective radiated power. It also supports dynamic system-level scenarios with algorithms for adaptive beamforming. Because

SystemVue includes MATLAB Script and supports baseband algorithm modeling in multiple languages, such as C++, SystemC, and VHDL/Verilog, SystemVue is an ideal platform to cross-validate phased array design information from RF, Baseband, and Test & Measurement teams. The Phased Array library is also compatible with SystemVue's many other reference libraries for 4G/5G, Radar, Satellite, and other modulation formats.

Various Geometries are being used for different applications when it comes to Phased Array Systems.



Objective

In typical R&D organization, many engineering disciplines are involved with Phased Array System design & development. While specific tools provide good enough capability to meet domain specific challenges, lot of gap exists in current tools to perform practical Phased Array System design and deal with subsystem’s interdependence issue. The Keysight SystemVue Phased Array solution allows Baseband, RF and Antenna teams to use a model-based engineering approach across disciplines.

This workshop intends to provide a detailed hands-on learning experience on Digital Beamforming, RF Phased Array analysis and Antenna Analysis to design Phased Array Systems (PAS) keeping many practical aspects of PAS into consideration which means designers can use Keysight SystemVue platform for their Phased Array System designs with higher degree of confidence.

Hands-On Labs

Session 1:
Digital Engineer's Perspective

Lab 1: Digital Beamforming with 3D Plots
Lab 2: Digital Beamforming with Digital Attenuator & Phase Shifter
Lab 3: Effects of Attenuator & Phase Shifter Quantization
Lab 4: AWGN effects

Session 2:
RF Engineer's Perspective

Lab 5: Phased Array design using RF Array Beamforming Kit
Lab 6: Quantization effect of Phase Shifters on Antenna Beam.
Lab 7: Phase Shifters and Attenuator S-Parameter (S2P) data effect on Antenna Beam
Lab 8: Amplifier Nonlinearity effect on Phased Array System
Lab 9: Phased Array Element Failure analysis using Monte Carlo

Session 3:
Antenna Engineer's Perspective

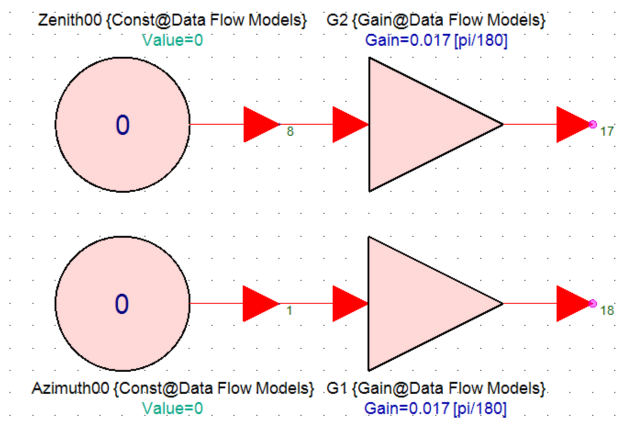
Lab 10: Antenna Active Loading
Lab 11: Importing EMPro/HFSS radiation pattern
Lab 12: Antenna Polarization Analysis

Lab 1: Digital Beamforming with 3D Plots

Objective of this lab is to understand Phased Array Beamforming components in SystemVue. A new real time 3D plotting capability is being introduce for Antenna Beamforming, Pattern and Pattern Measurement. In this Lab, a simple Rectangular Array of size 8X8 is defined and we will plot and measure Antenna Far Field Pattern for this array.

Step1: Create a new Workspace with a name Lab1_Beamformer.

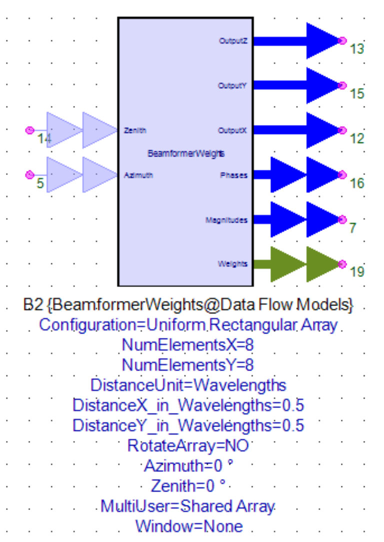
Step2: From Algorithm Library, place “Const” & “Gain” components onto schematic page and create 2 inputs as shown here representing “Theta” & “Phi” i.e. Zenith and Azimuth values.



Step3: Double click on Const component and rename them as Zenith00 and Azimuth00

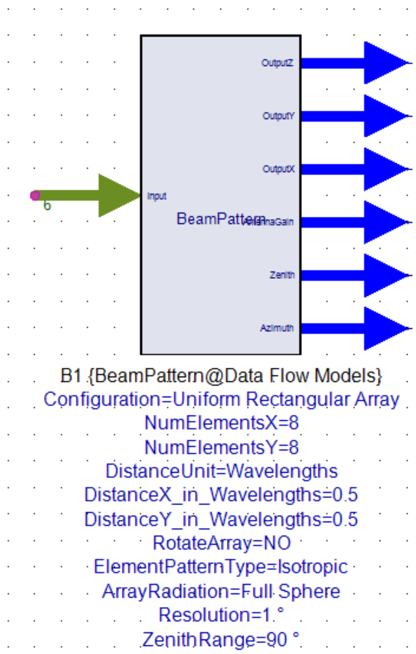
Step4: Double click on Gain component and define Gain = pi/180

Step5: From Algorithm Design library, place BeamformerWeights onto schematic. Double click and specify the parameters as shown below. Beamformer block can provide the Weight coefficients (can be used for digital control), Phase and Magnitude coefficients (can be used for controlling Digital Attenuator & Phase Shifter)



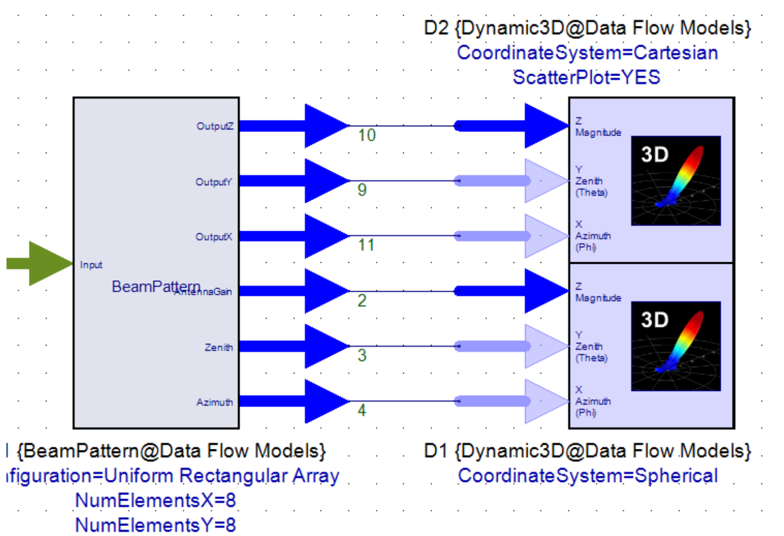
Name	Value	Units
Configuration	2:Uniform Rectangular Array	
NumElementsX	8	
NumElementsY	8	
DistanceUnit	1:Wavelengths	
DistanceX_in_Wavelengths	0.5	
DistanceY_in_Wavelengths	0.5	
RotateArray	0:NO	
Azimuth	[0]	deg
Zenith	[0]	deg
MultiUser	0:Shared Array	
Window	0:None	

Step6: From Algorithm Design library, place BeamPattern component onto schematic. Double click and specify the parameters as shown below. BeamPattern block synthesizes the Antenna Beam based on the input coefficients.

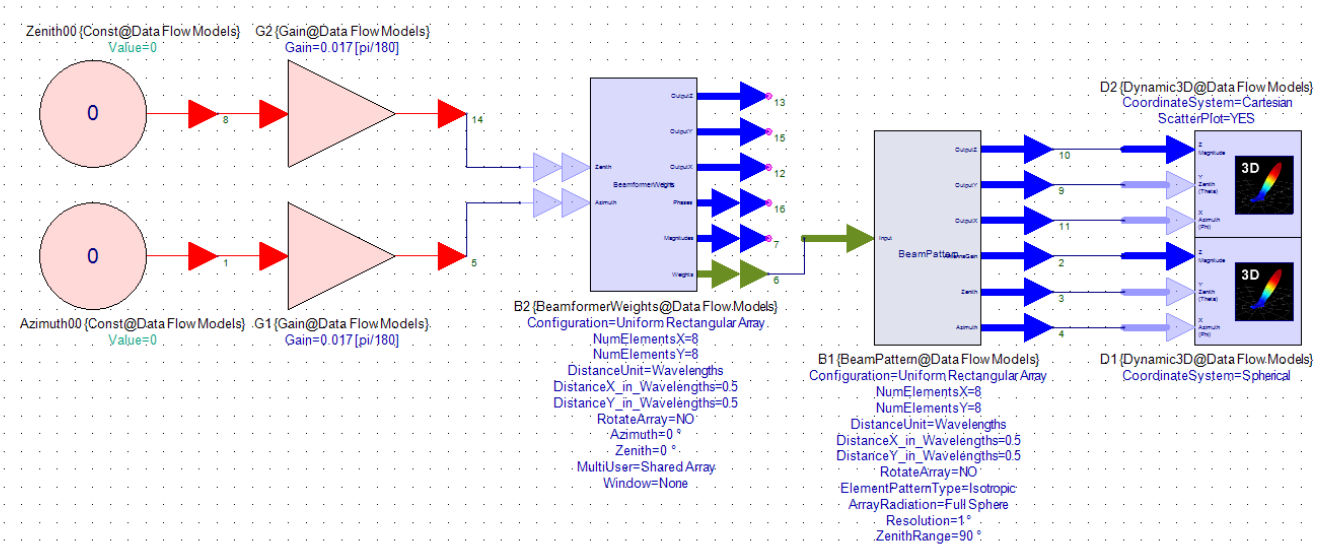


Name	Value	Units
Configuration	2:Uniform Rectangular Array	
NumElementsX	8	
NumElementsY	8	
DistanceUnit	1:Wavelengths	
DistanceX_in_Wavelengths	0.5	
DistanceY_in_Wavelengths	0.5	
RotateArray	0:NO	
ElementPatternType	0:Isotropic	
ArrayRadiation	0:Full Sphere	
Resolution	1	deg
ZenithRange	90	deg

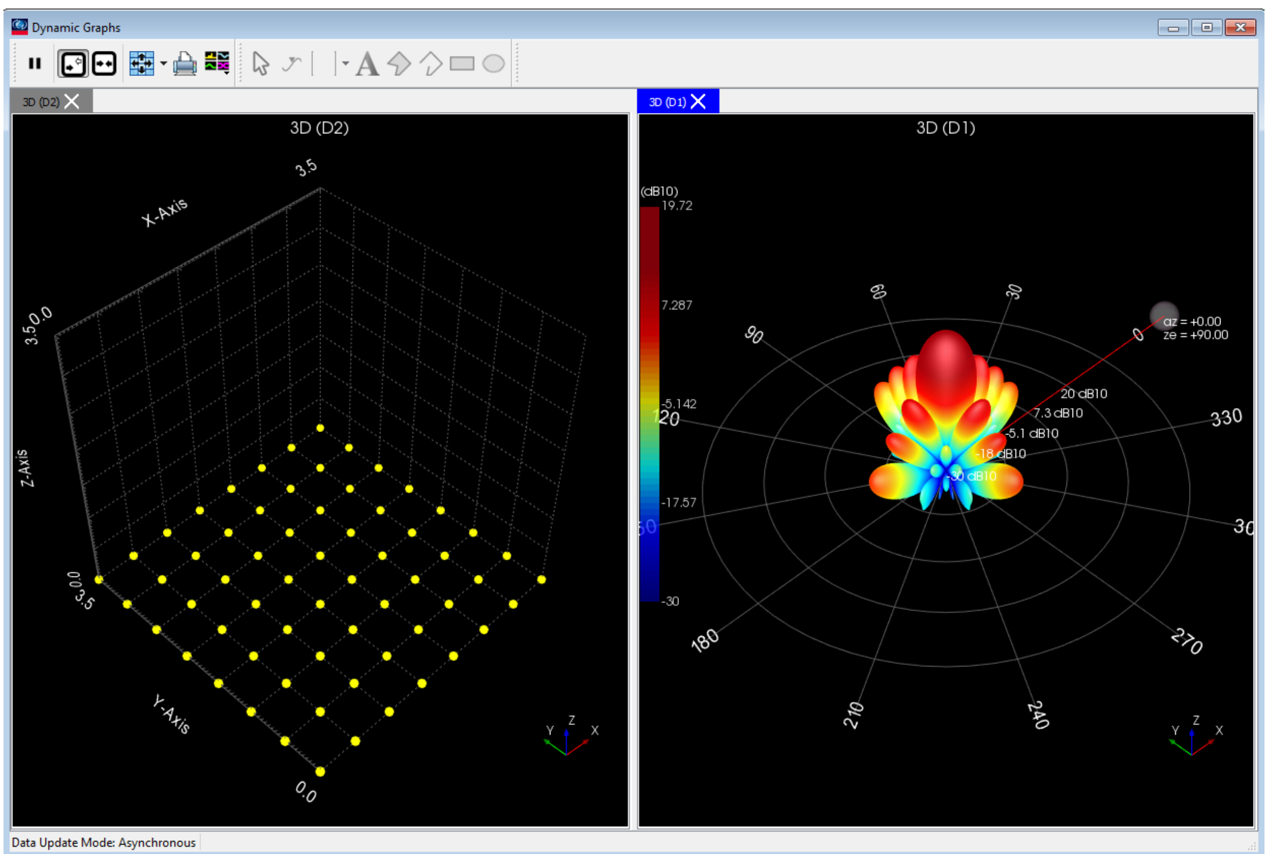
Step7: Connect the Dynamic3D components to BeamPattern output as shown here. 1st Dynamic3D (Spherical) plot component will be connected to AntennaGain, Zenith & Azimuth output pins.2nd Dynamic3D (Cartesian) plot component will be connected to OutputZ, OutputY & OutputX output pins.



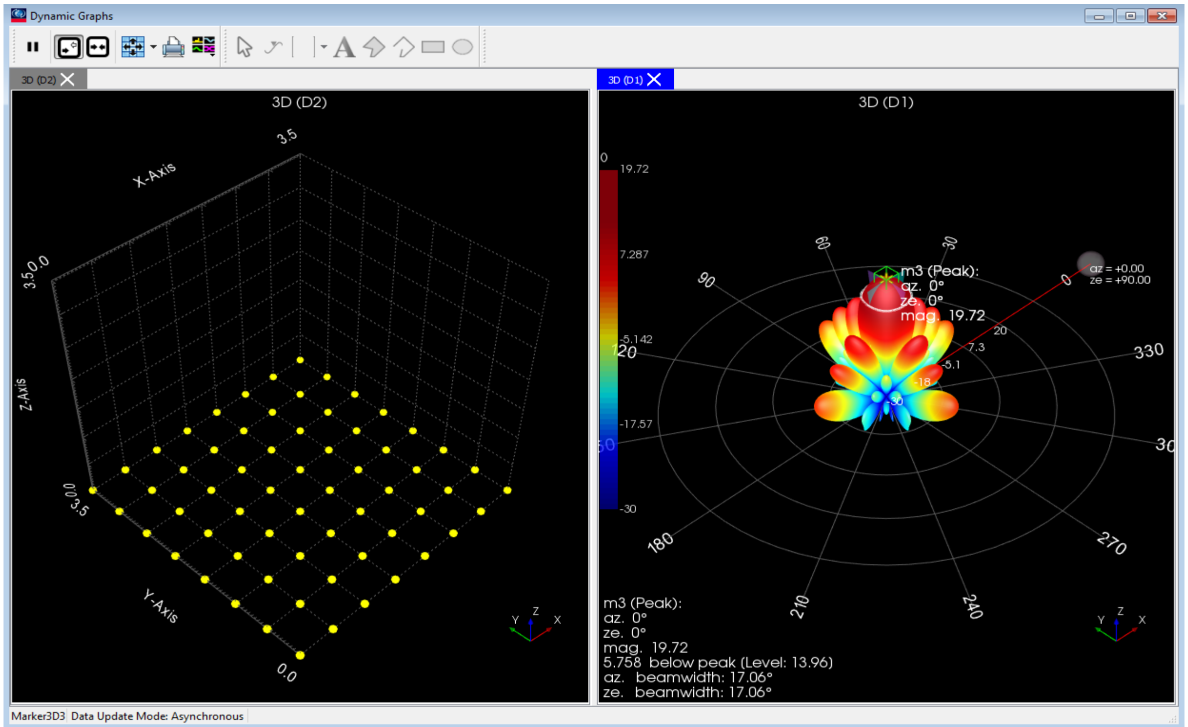
Note: Once connected, schematic design should appear as below



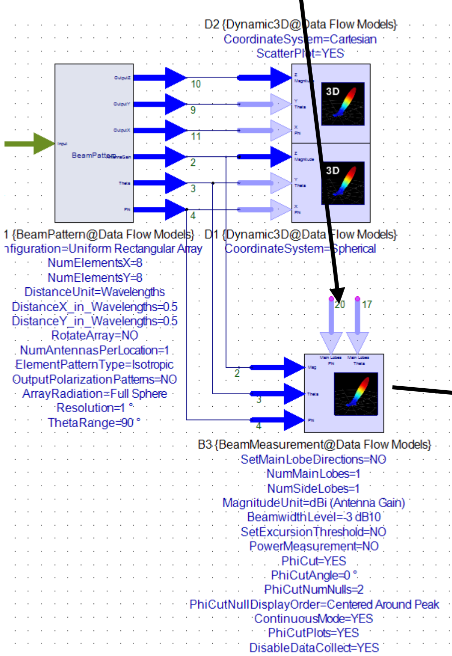
Step8: Run simulation and observe Dynamic 3D plot as shown here. Right Click on Antenna Beam, select Markers->Peak and note Azimuth and Zenith which should be 0, 0.



Step9: Right click on 3D beam again and select Beamwidth->3dB Below to notice Az and Ze Beamwidth etc. Stop simulation when satisfied…..

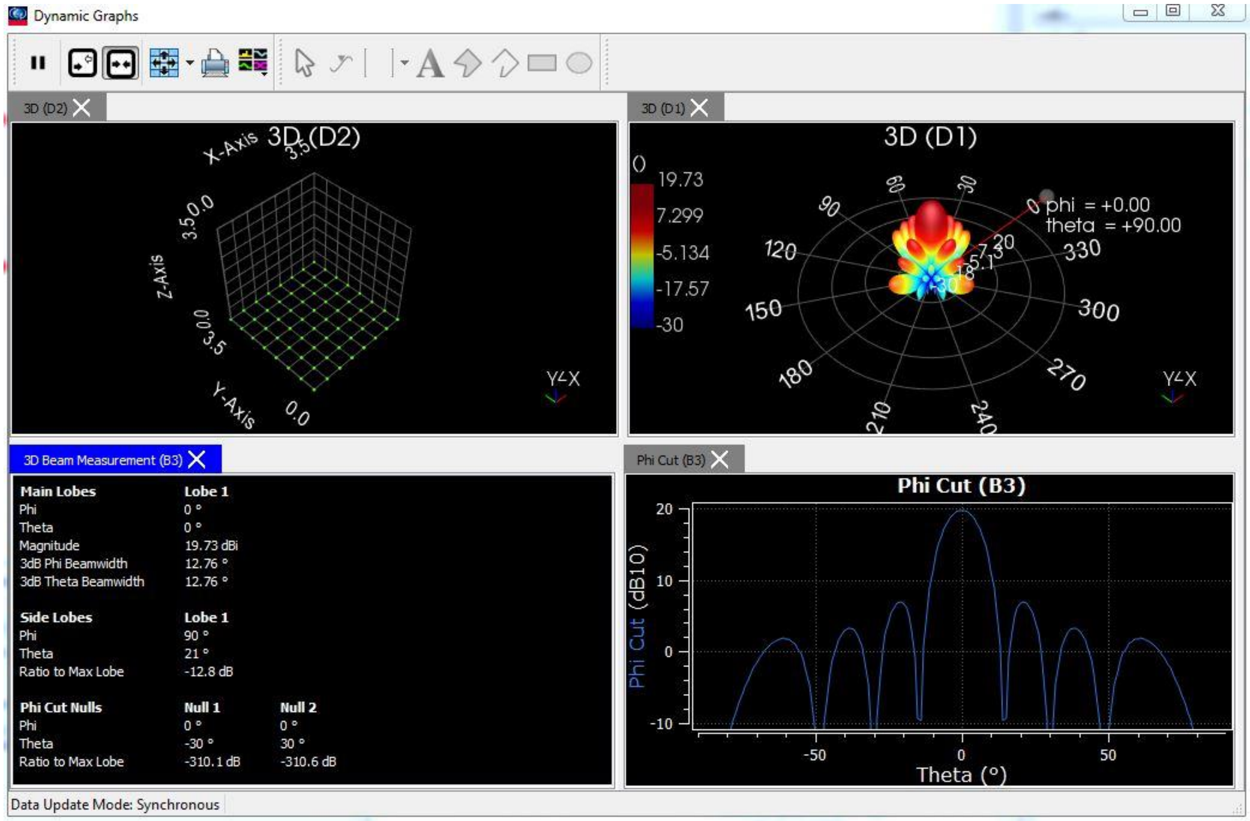


Step 10: Insert BeamMeasurement component from Algorithm Design library and connect it to AntGain, Theta and Phi output of BeamPattern component and set **PhiCut = Yes**



Name	Value	Units
SetMainLobeDirections	0:NO	
NumMainLobes	1	
NumSideLobes	1	
MagnitudeUnit	0:dBi (Antenna Gain)	
BeamwidthLevel	-3	dB
SetExcursionThreshold	0:NO	
PowerMeasurement	0:NO	
PhiCut	1:YES	
PhiCutAngle	0	deg
PhiCutNumNulls	2	
PhiCutNullDisplayOrder	0:Centered Around Peak	
ContinuousMode	1:YES	
PhiCutPlots	1:YES	
DisableDataCollect	1:YES	

Step 11: Run Simulation and observe all the key beam measurements and 2D Phi Cut as shown below (adjust the plots in the Results window as needed). Set the Y-axis of the 2D cut to +20 to -10 for proper 2D cut display.



Step12: Stop Simulation once satisfied. **Save the Workspace**

Step 13: Change Azimuth and Zenith angles and re-run simulations**.** Change Array Size in BeamformerWeights and BeamPattern and re-run simulations to explore different options.

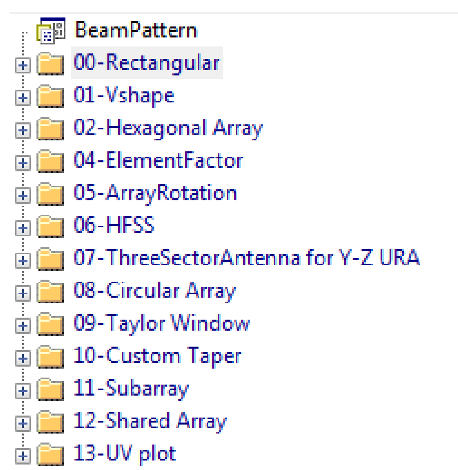
Optional Exercise to Explore Different Antenna Geometries…

- Click on **Help->Open Example->Beamforming**

  and select to open **BeamPattern** workspace

- Notice folders for different kind of beam patterns implemented in this example

- Select the ones which you wish to see and simulate the respective design to observe the beam pattern

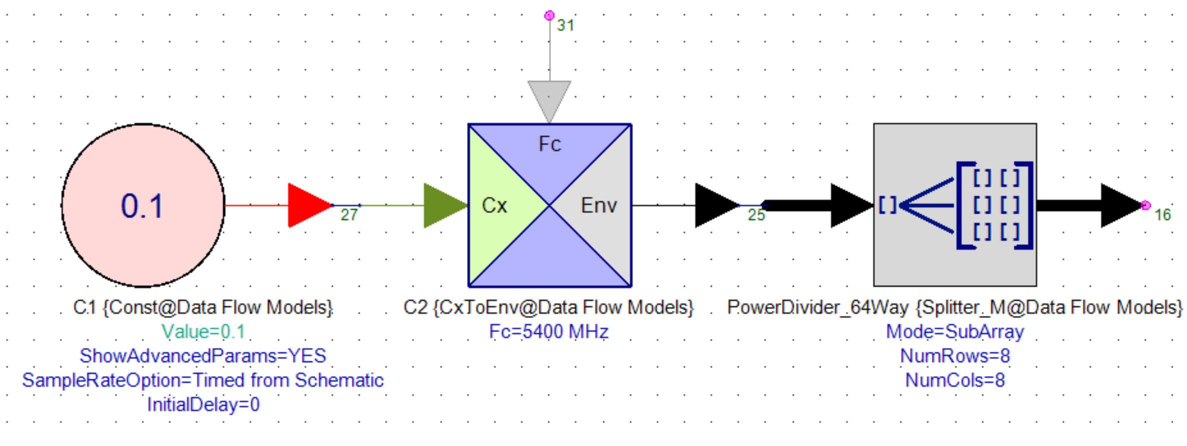


Lab 2: Digital Beamforming with Digital Attenuator & Phase Shifter

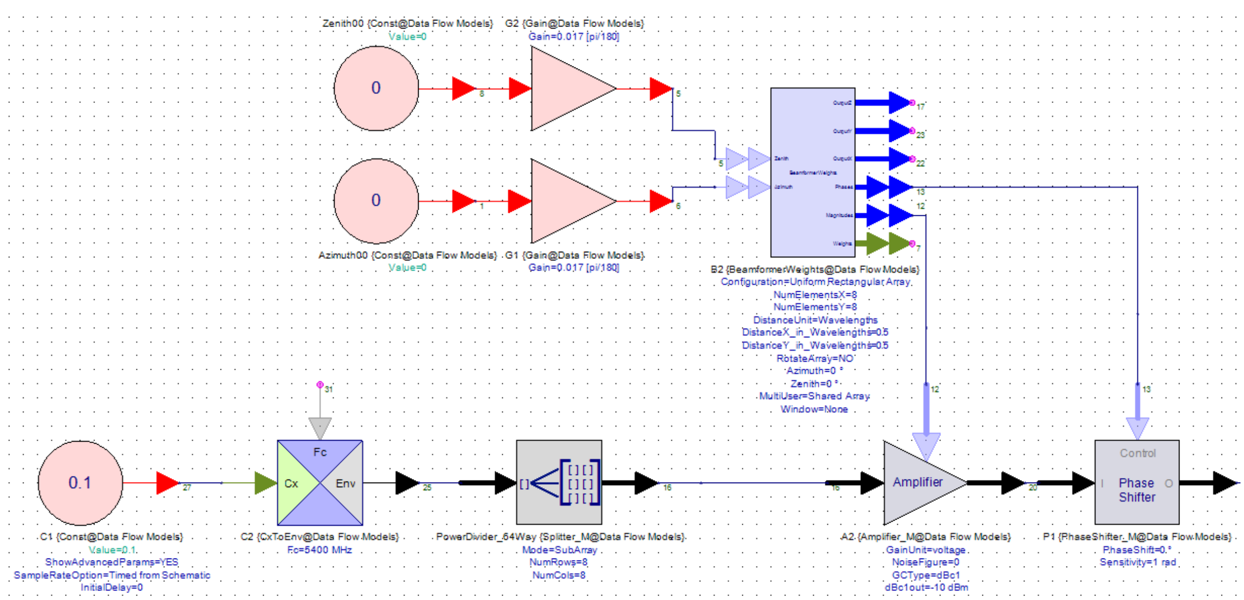
Objective of this lab is to understand the effects of Digital Attenuator and Digital Phase Shifter on the Phased Array Beam Forming. We will look at the Far Field pattern to see impact of these components on eventual output of the Phased Array System. RF modeling of these two critical components is kept to the simplest possible level so that it is convenient for Digital Engineer navigate his way without going too deep into RF modeling aspects.

Step1: Open Lab1_Beamformer workspace and use File->SaveAs to save it with name **Lab2_Beamformer_with_RF**

Step2: From Algorithm Design library, place **Const, CxToEnv, Splitter_M, Amplifier_M and PhaseShifter_M** components on schematic and specify the parameters as shown here

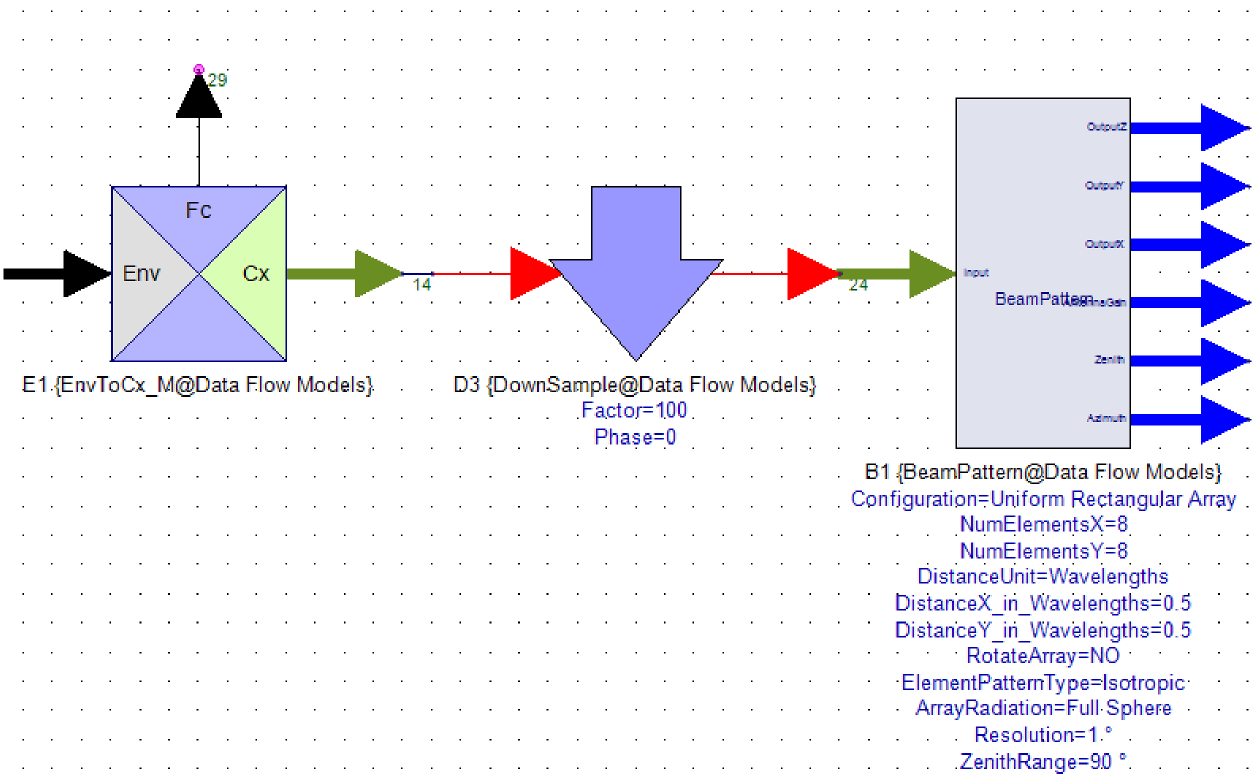


Step3: Disconnect the output of BeamFormerWeights which connects to BeamPattern component and connect the Magnitudes & Phases Output pins of BeamFormerWeights to the Amplifier_M & PhaseShifter_M control Pins respectively as shown here

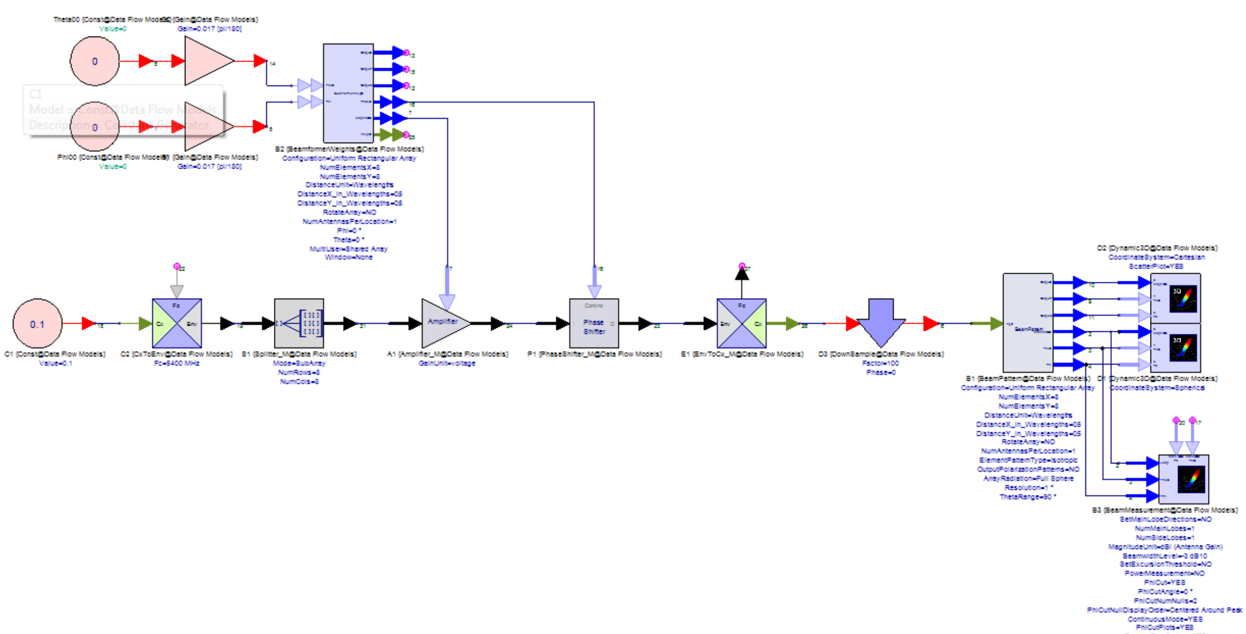


Double click PhaseShifter_M component and define Sensitivity as 1 rad

Step4: Place EnvToCx_M and DownSample from Algorithm Design library onto the schematic. Define Downsample Factor = 100 and connect the Downsampler output to BeamPattern input as shown here:

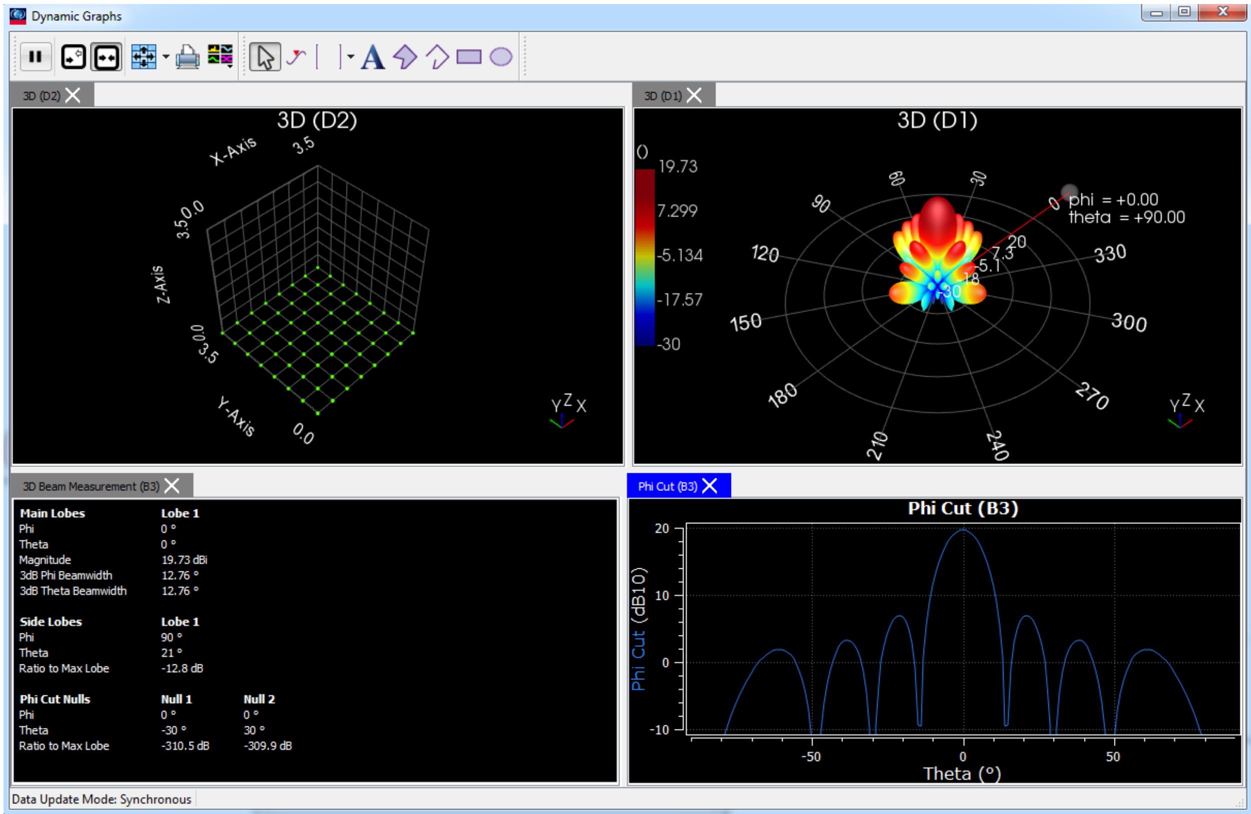


Completed schematic will appear as shown below:



10

Step5: Run Simulation and observe Antenna Pattern as well as 2D cut with related measurements and it appears to be pretty similar to the earlier results (as expected).

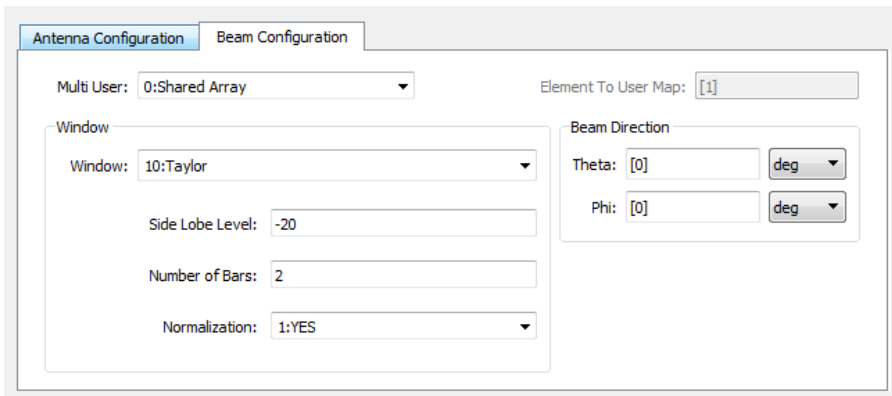


Lab 3: Effects of Attenuator & Phase Shifter Quantization

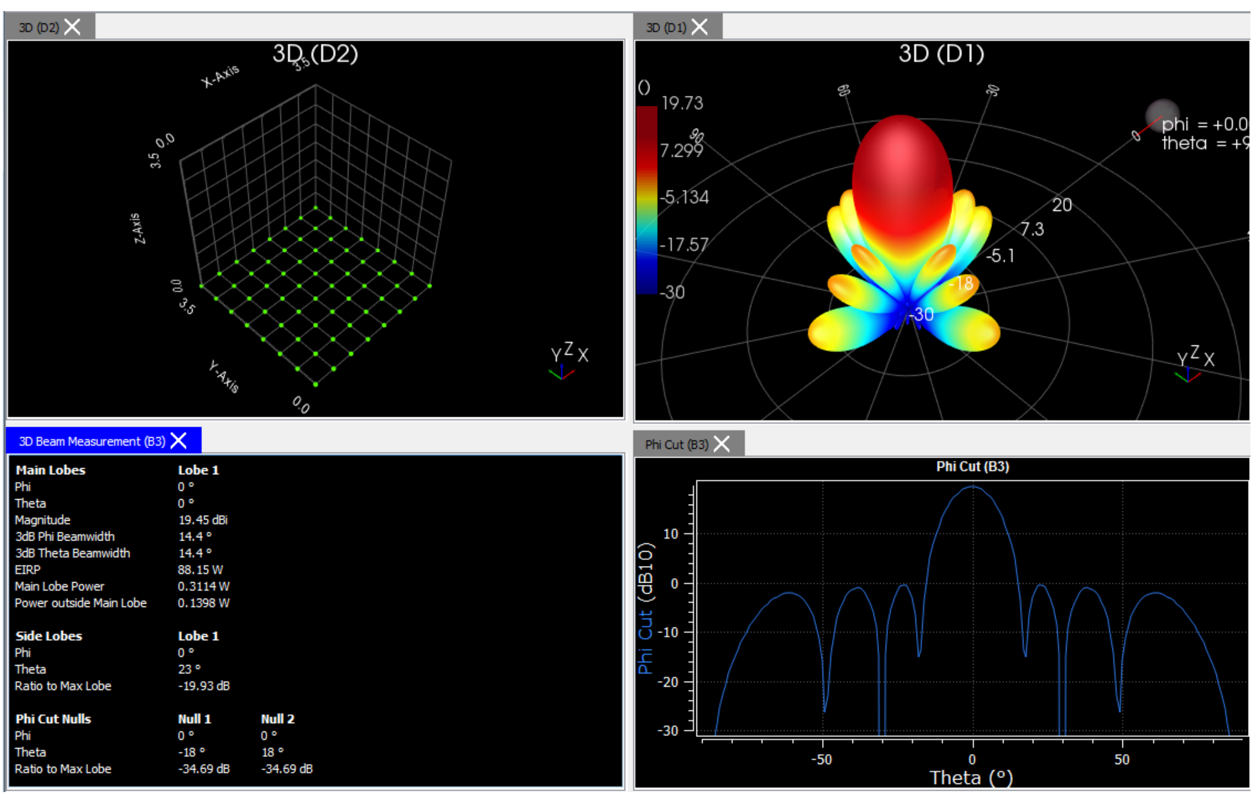
Objective of this lab is to model little more realistic specification of Digital Attenuator and Digital Phase Shifter on the Phased Array Beam Forming. Again, we will keep the RF modeling aspects to the simplest possible level to make Digital Engineer comfortable but still do his job more effectively.

Step1: Save the Lab2….. workspace with name **Lab3_Beamformer_withQuant**

Step2: Double click the Beamformer, select Window as Taylor from Beam Configuration tab. Set parameters as shown below



Step3: Run Simulation and observe the results as shown below. Notice Side lobes output in the 3D Beam Measurement panel



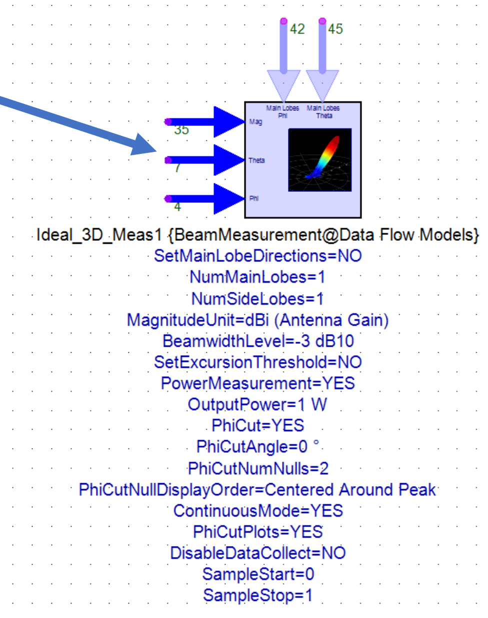
Step4: Double click on Beam Measurement component in both the branches and setup parameters as shown here:

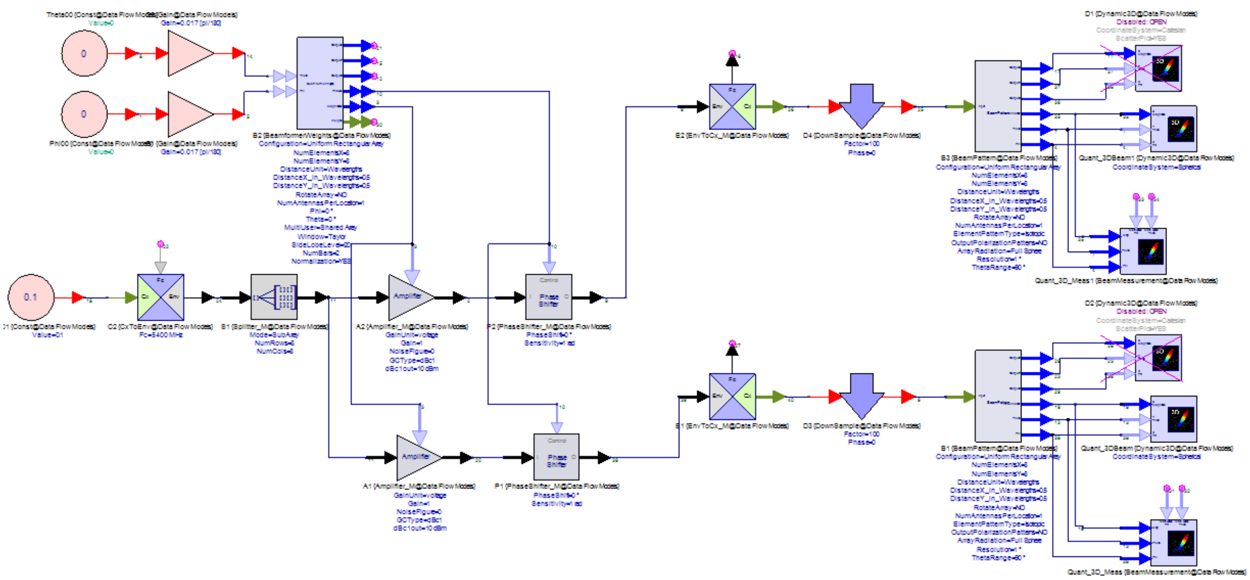
Step5: Copy/Paste Attenuator and Phase Shifter components and connect it in parallel to the existing line up.

Step6: Copy/Paste rest of the Beam rendering component to create parallel chain as shown below.

Step7: Disable Cartesian 3D Plots that is connected to Output X,Y,Z of BeamPattern component.

.





Step8: Double click on Attenuator & Phase Shifter components (parallel chain) and specify characteristic as shown below. Here we are using 6-bit attenuator & phase shifter with some amplitude and phase error which will be there in the real-world components

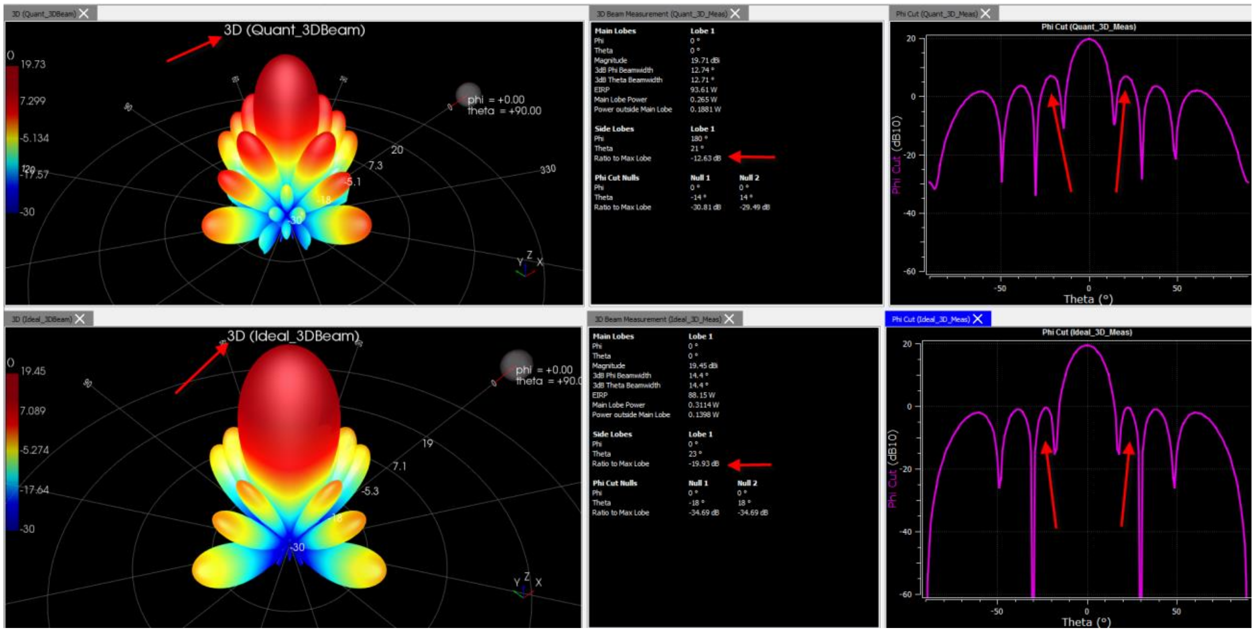
Name	Value	Units	Default
GainUnit	0:voltage		0:voltage
Gain	[1.0]		[1.0]
Quantization	1:Number of Bits (Uniform)		0:NO
NumBits	6		6
StepSize	0.1	dB	0.5 dB
MaxGain	10	dB	10 dB
GainError	1:Normal		0:None
StdDev	0.1	dB	0.5 dB
NoiseFigure	0		0
GCType	2:dBc1		0:none
dBc1out	10	dBm	0.01 W
RefR	50	Ohm	50 Ω

Name	Value	Units
PhaseShift	[0.0]	deg
InsertionLoss	0	dB
Quantization	1:Number of Bits (Uniform)	
NumBits	6	
PhaseShiftError	1:Normal	
StdDev	1	deg
Sensitivity	1	rad
HilbertFilterLength	64	

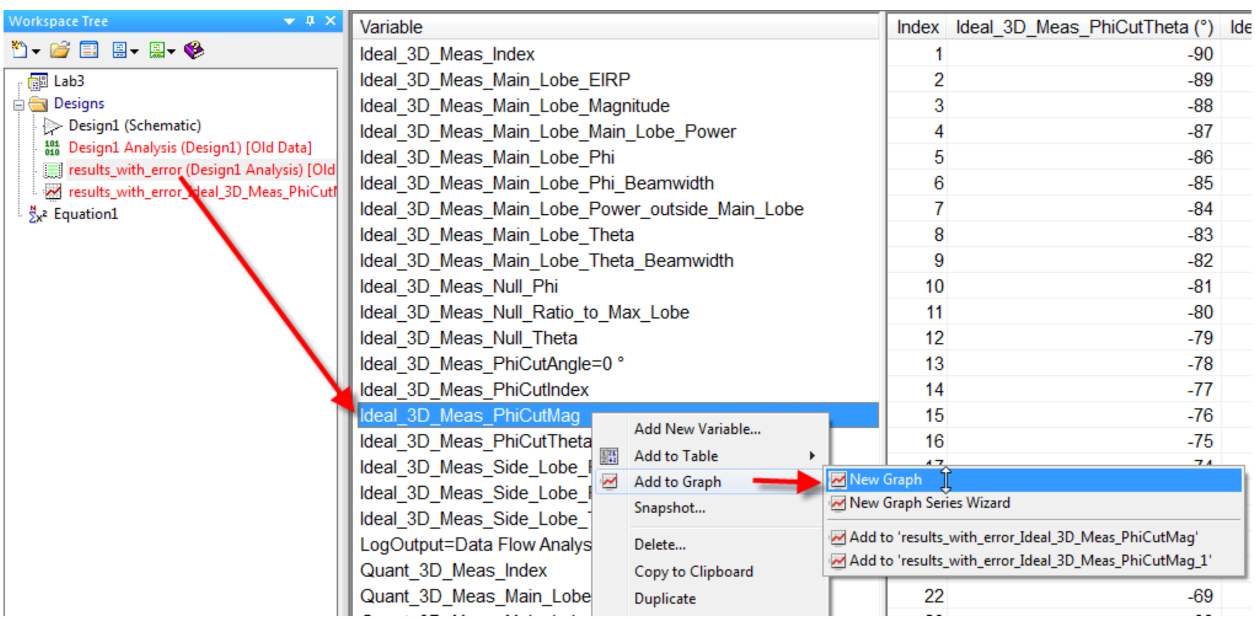
Step9: Rename beam measurement components connected to the respective branches to identify the results properly e.g. use prefix of **Ideal** and **Quant**

Step10: Run Simulation and arrange plots side by side for quick comparison. Notice the degradation in the side lobe levels between 2 cases which is pretty clear from the measurements.

Also, we can see that 3dB Beamwidth gets affected and it is different in both the cases.

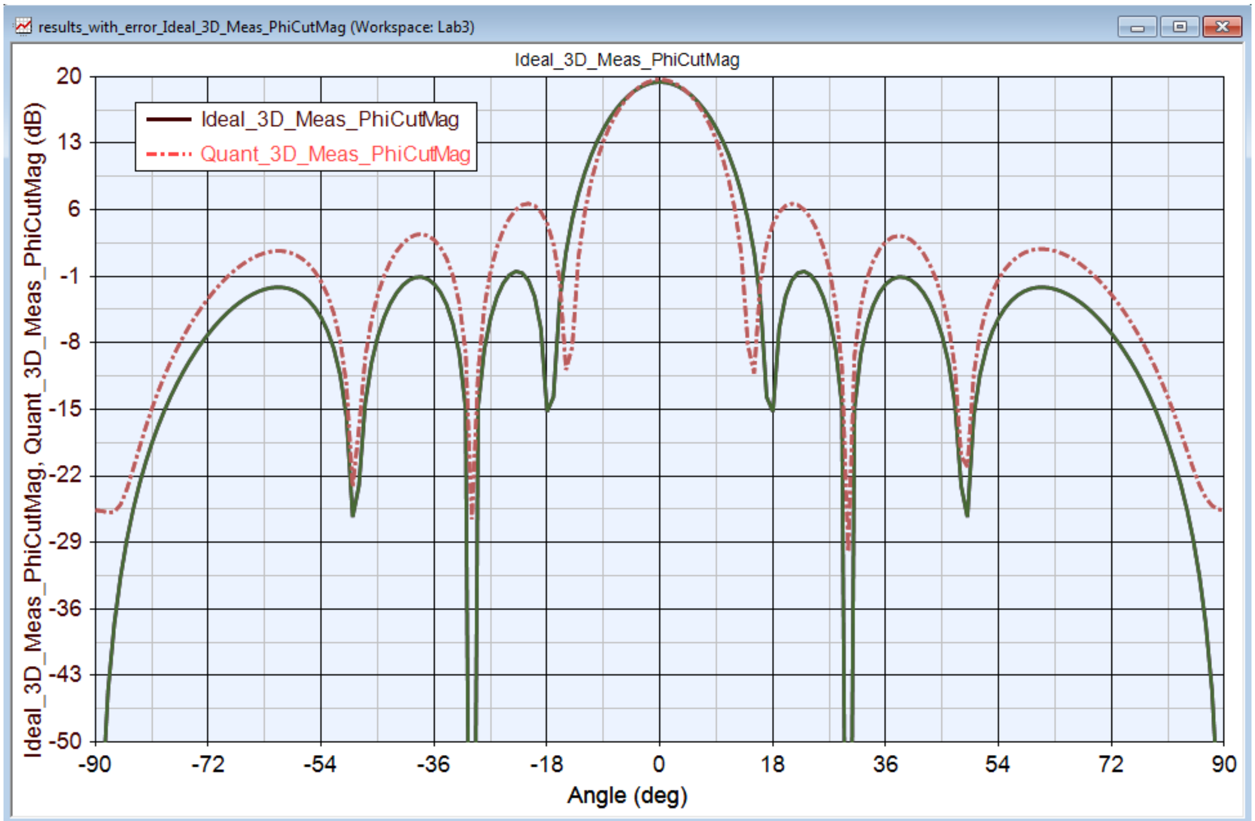


Step11: To get clearer picture and perform results comparison, double click to open the dataset from the workspace tree. Select <prefix…..**PhiCutMag**> right click and select Add to Graph->New Graph



Repeat the same for other PhiCutMag result and this time select the graph name which appears as the 3rd option in the list (name of the graph can be seen in the workspace tree) so that traces get overlayed for easy comparison as shown on the next page

Plot below shows Phi Cut response and we can clearly see the rise in side lobe levels and difference in the beam width due to quantization and Mag/Phase errors



Lab 4: AWGN effects

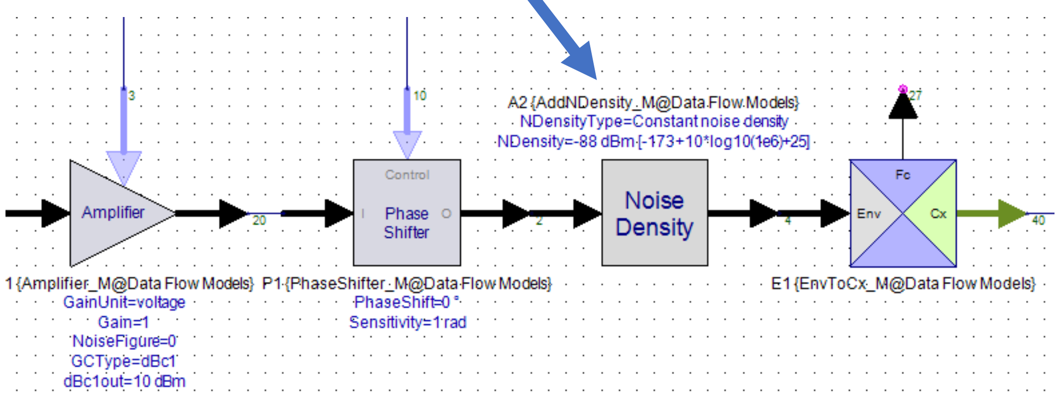
Objective of this lab is to understand the effects caused by AWGN channel on our emitted Antenna Beam.

Step1: Save Lab3… with name **Lab4_Beamforming_with_AWGN**

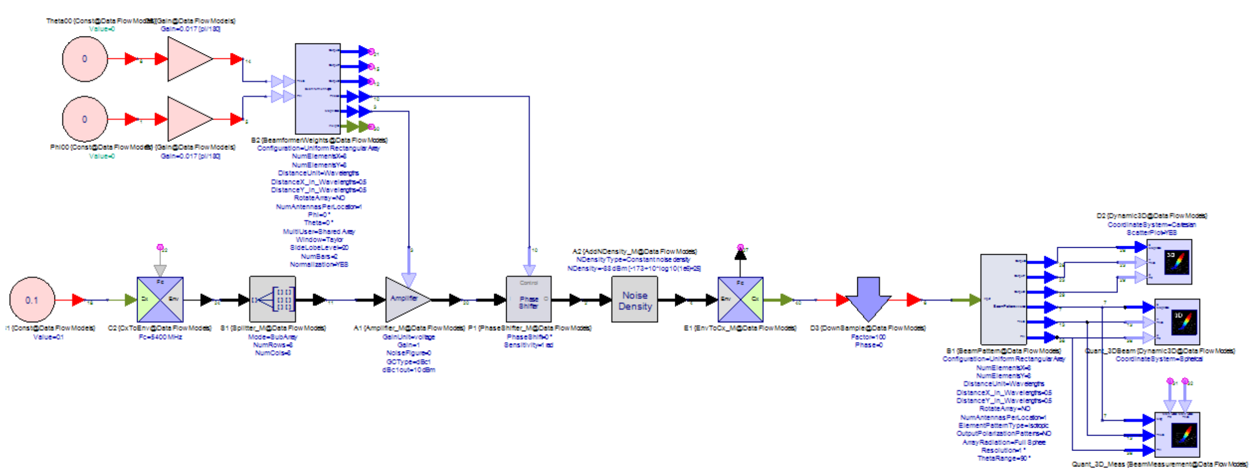
Step2: Disable **or** delete all the components connected to ideal Attenuator & Phase Shifter

Step3: Insert **AddNDensity_M** component at the output of PhaseShifter and specify its parameter as shown here

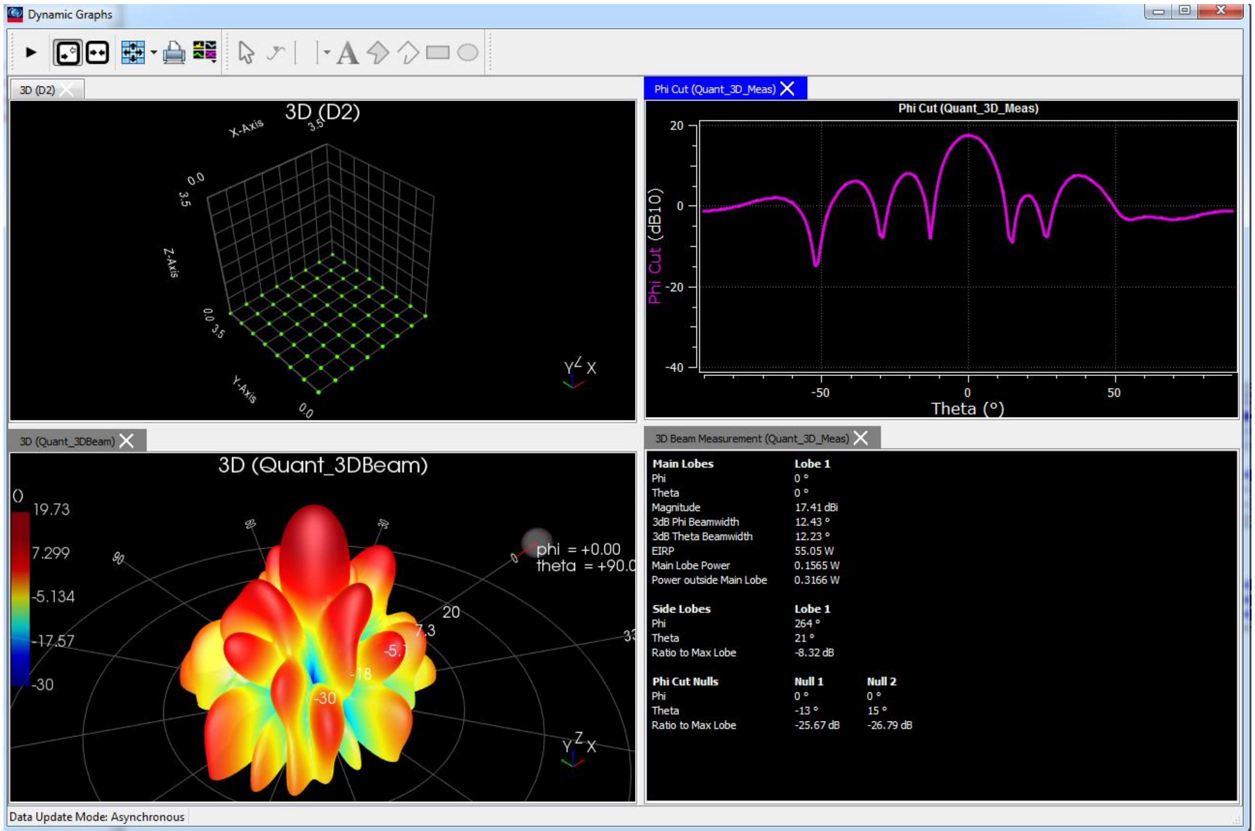
Name	Value	Units
NDensityType	0:Constant noise density	
NDensity	-173+(10*log10(1e6))+25	dBm
RefR	50	Ohm



Completed system should appear similar to the one shown below (here ideal components used in earlier lab is deleted)



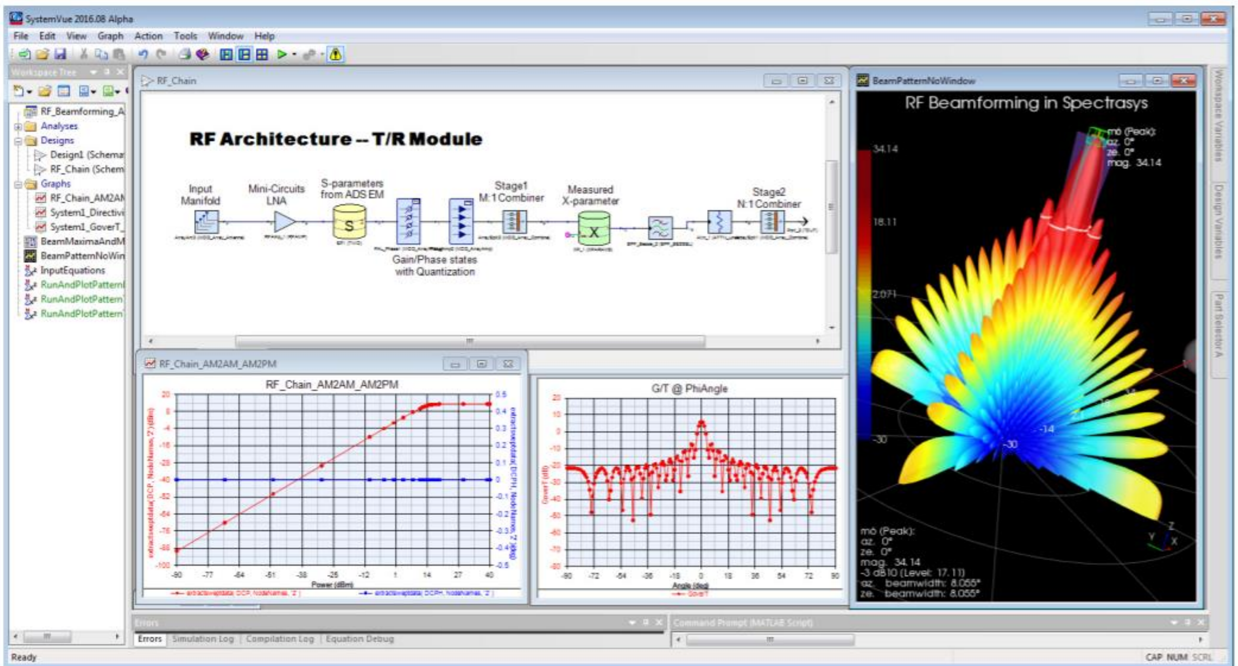
Step4: Run Simulation to see the impact of AWGN Noise onto Antenna Radiation Pattern and degradation due to the same



RF Phased Array Analysis

Key Highlights:

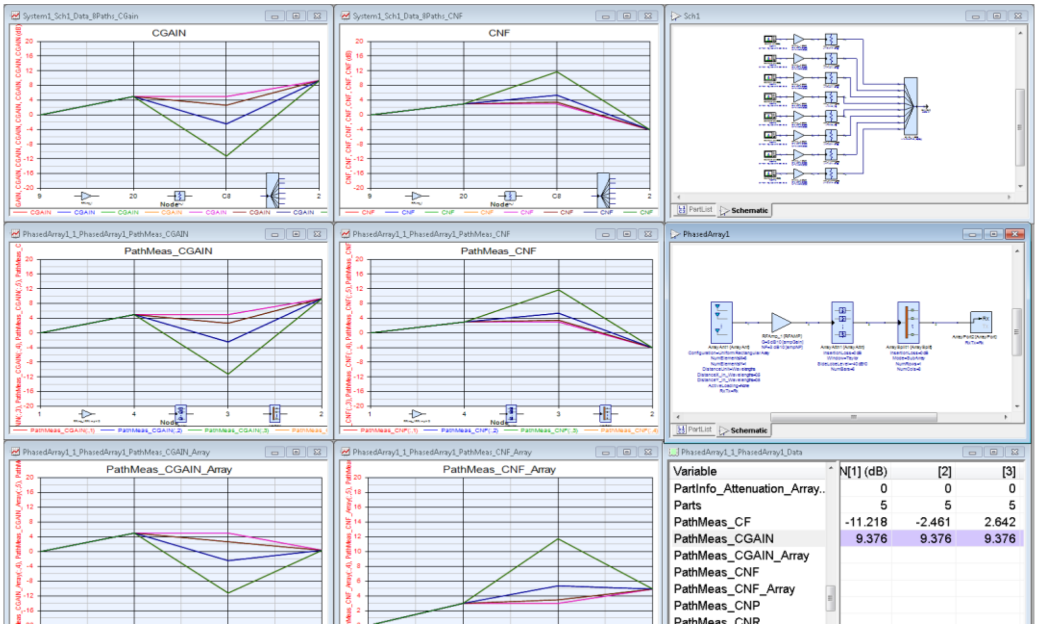
- RF array analysis enhances Spectrasys simulator in Keysight SystemVue to work with phased array system. Replaces ad hoc spreadsheets with nonlinear RF analysis for N-element arrays used in RF beamforming, with multiple levels of splitters/combiners, gain/phase states, and impairments from nonlinear, mismatch, and noise effects.



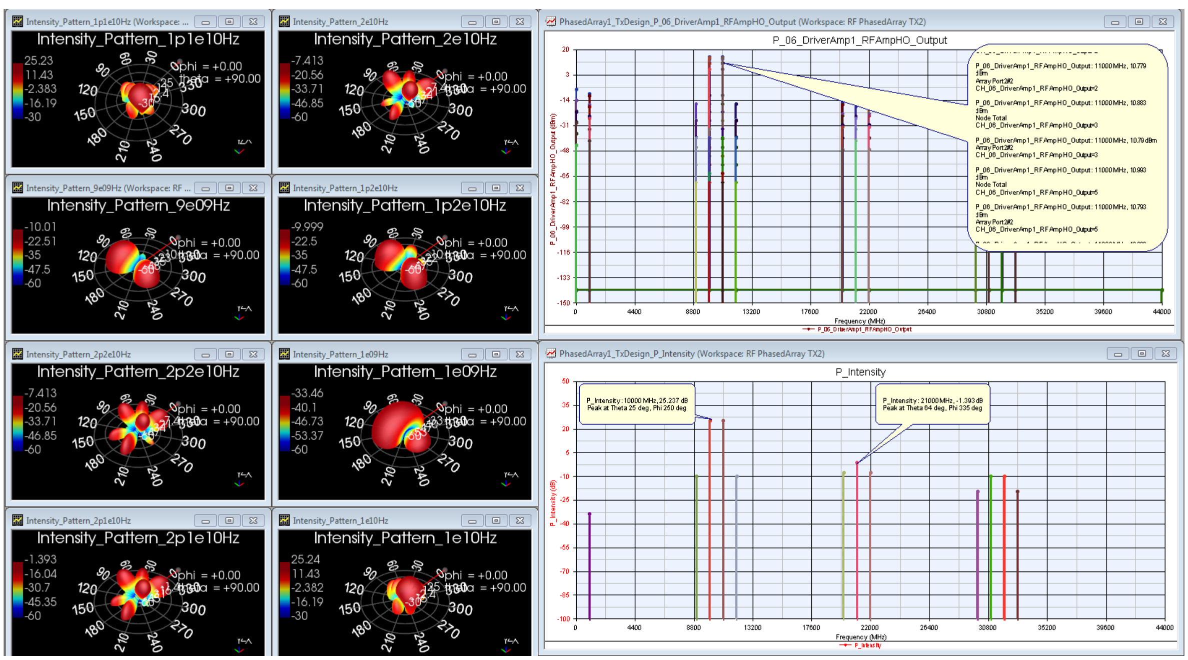
- Using S-, X- and Sys-parameters of phase-shifters, attenuators, amplifiers and mixers for fast hardware implementation. – Antenna element failure analysis - Monte Carlo or user-specified element failures enables robust mission critical design.

- Antenna element failure analysis - Monte Carlo or user-specified element failures enables robust mission critical design.

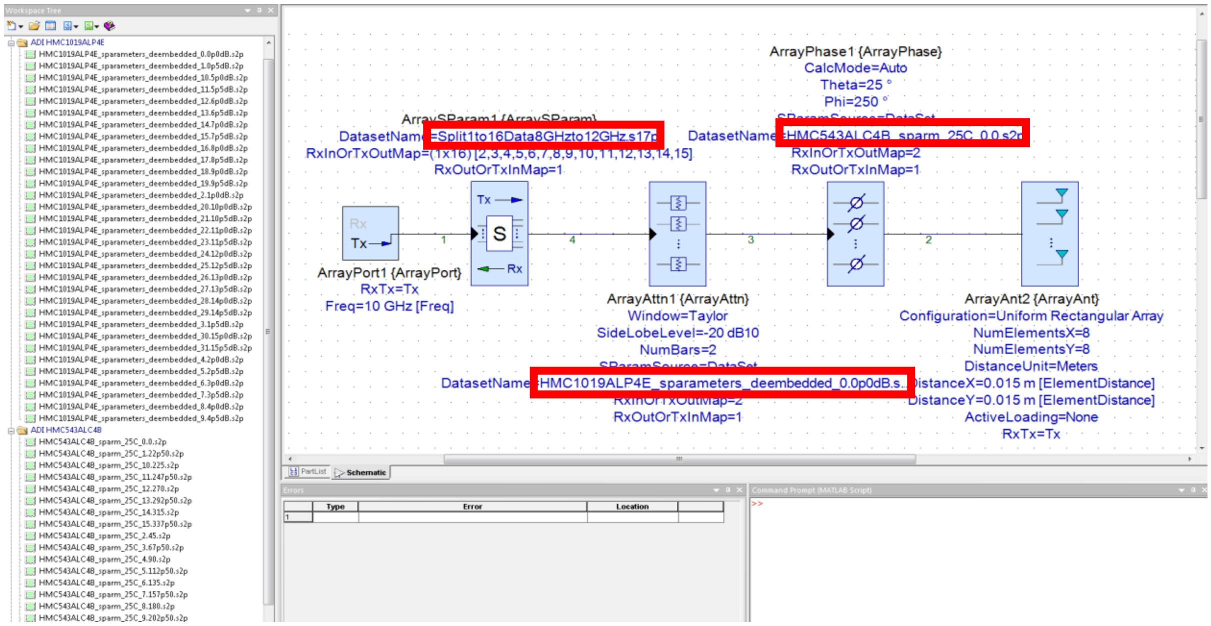
- Individual Path Measurement in RF Phased Array Analysis.



- Using RF Analysis Pattern and Spectrum for Intermods and Harmonics can be plotted.



- S-Parameter model for Phase Shifter and Anttenuators can be used in Phased Array Analysis.



Lab 5: RF Phased Array Design

Objective of this lab is to design 8X8 Phased Array Transmitter using RF Phased Array models which provides much better RF fidelity/characteristics as needed by a RF System engineer for very accurate RF chain analysis. We will perform the Frequency domain simulation to perform RF system budget analysis along individual array paths, plotting 3D Far Field Pattern and 2D Cuts.

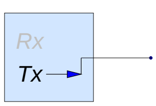
Step1: Open a new workspace and save it as Lab5_RF_Phased_Array

Step2: Double click on **Equation1** under the workspace tree and define the parameters as below:

(***take care of the syntax of the variables you define here and use the same while using these variables in the RF system components later***)

RefFreq = 10e9;	***% Frequency used for setting element spacing***
LightSpeed = 299792458;	***% m/sec***
% Half wavelength based on RefFreq	
ElementDistance = 0.5*LightSpeed / RefFreq;	
% Operating frequency	
Freq = tune(10);	***% in GHz***

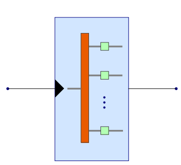
Step3: From RF Design Library bring the **Phased Array RxPort/TxSource** (search with name Array) and place it on the Schematic. This will act as a Source to our phased array system. Define the parameters as below, ***take care of the syntax of the variables***….



ArrayPort1 {ArrayPort}
RxTx=Tx
Freq=10 GHz [Freq]
Phase=0 °
MainCarrierIndex=1

Name	Value	Units	Default	Ise Defaul	Tune	Show
RxTx	2:Tx		1:Rx			✓
Freq	Freq	GHz	[1000] MHz			✓
TxPwrIn	0	(dBm)	[0] dBm	✓		
Phase	[0]	(deg)	[0] °	✓		✓
MainCarrierIndex	1		1	✓		✓
Z0	50	(Ohm)	50 Ω	✓		
ZFreq	0	Hz	0 Hz	✓		
PORT	1		1			
Ta		°C	°C	✓		

Step4: We will use an Array Splitter to divide the source and feed to each array path. From RF Design Library bring the Phase Array Combiner/Splitter Model. It has default setup for 8X8 so no changes needed as of now. Parameter set will be as shown below:

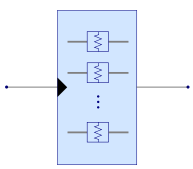

ArraySplit1 {ArraySplit}
InsertionLoss=1 dB
Mode=SubArray
NumRows=8
NumCols=8

Name	Value	Units	Default	Ise Defaul	Tune	Show
InsertionLoss	1	(dB)	1 dB	✓		✓
Mode	0:SubArray		0:SubArray	✓		✓
NumRows	8		8	✓		✓
NumCols	8		8	✓		✓
Zref	50	(Ohm)	50 Ω	✓		
Zin	50	(Ohm)	50 Ω	✓		
Zout	50	(Ohm)	50 Ω	✓		
VSupply	0	V	0 V	✓		
ISupply	0	mA	0 mA	✓		
Ta		°C	°C	✓		

Step5: Place Array Attenuator model from RF Design Library. In this model, we can define the Windowing parameter and for our design we will define Window = Taylor. We have not used Quantization in this case, however you can define the Quantization and define the number of bits pretty much as we did in earlier lab exercise.

Note: *When Window = Taylor, the SideLobeLevel and NumBars parameters are used to generate Taylor window. The generated Taylor window (when applied over a uniformly illuminated array) will produce a beam pattern that will have NumBars sidelobes approximately at SideLobeLevel dB below the peak.* For more information please refer Model Help.

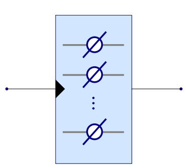
Define the parameters for attenuator as shown below:



ArrayAttn1 {ArrayAttn}
Window=Taylor
SideLobeLevel=-20 dB10
NumBars=2
InsertionLoss=0 dB

Name	Value	Units	Default	Ise Defaul	Tune	Show
Window	10:Taylor		0:None	☐	☐	☑
SideLobeLevel	-20	dB	-20 dB	☐	☐	☑
NumBars	2		2	☐	☐	☑
Quantization	0:NO		0:NO	☐	☐	☐
InsertionLoss	0	(dB)	1 dB	☐	☐	☑
Error	0:None		0:None	☑	☐	☐
Zin	50	(Ohm)	50 Ω	☑	☐	☐
Zout	50	(Ohm)	50 Ω	☑	☐	☐
Zref	50	(Ohm)	50 Ω	☑	☐	☐
VSupply	0	V	0 V	☑	☐	☐
ISupply	0	mA	0 mA	☑	☐	☐
Ta		°C	°C	☑	☐	☐

Step6: Bring Array Phase Shifter model from RF Design Library. The Array Phase model is used to apply a phase shift to the individual signals in a Phased Array Antenna system. Define the parameters as below. Again Quantization can be defined as needed (not done in this lab)

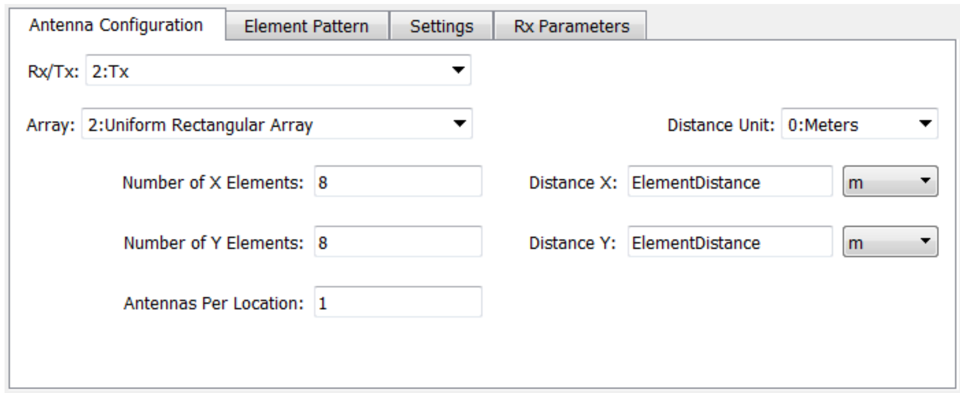


ArrayPhase1 {ArrayPhase}
CalcMode=Auto
Theta=25 °
Phi=250 °
InsertionLoss=0 dB

Name	Value	Units	Default	Ise Defaul	Tune	Show
CalcMode	0:Auto		0:Auto	☑	☐	☑
Theta	25	deg	0 °	☐	☐	☑
Phi	250	deg	0 °	☐	☐	☑
FrequencyOption	0:From schematic		0:From sc...	☑	☐	☑
Frequency	1000	(MHz)	1000 MHz	☑	☐	☑
Quantization	0:NO		0:NO	☐	☐	☐
InsertionLoss	0	(dB)	1 dB	☐	☐	☑
Error	0:None		0:None	☑	☐	☐
Zin	50	(Ohm)	50 Ω	☑	☐	☐
Zout	50	(Ohm)	50 Ω	☑	☐	☐
Zref	50	(Ohm)	50 Ω	☑	☐	☐
VSupply	0	V	0 V	☑	☐	☐
ISupply	0	mA	0 mA	☑	☐	☐
Ta		°C	°C	☑	☐	☐

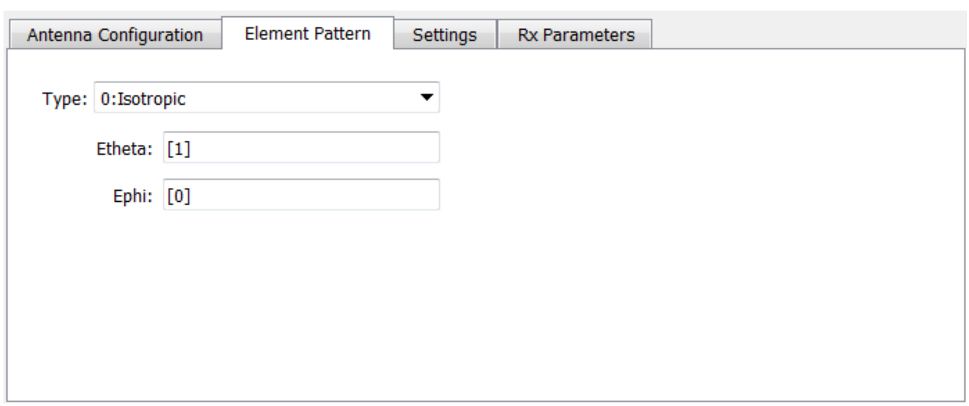
Step7: Bring Array Antenna model from RF Design Library. Double click and define the parameters in each tab.

1. Under Antenna Configuration define the parameters as below:

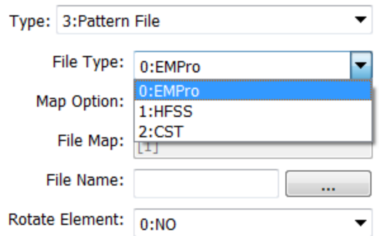


- We have defined the mode as Tx.
- Array geometry as Uniform Rectangular Array. We can define any one of Linear, Circular, Hexagonal or Custom etc.
- We have defined the number of Elements in X and Y direction.
- In Step2, we have defined Element Distance and same shall be used here to define Distance X and Distance Y which is set as half-wavelength

2. Under Element Pattern define the parameters as below:

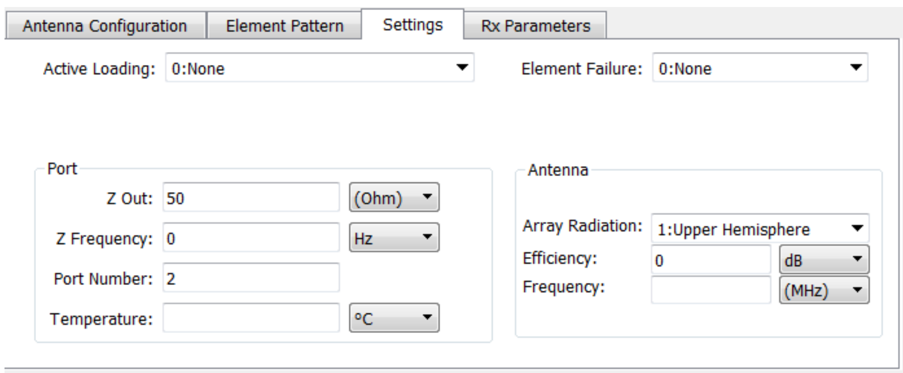


- Under Type we define the Antenna Radiation Pattern.
- We have defined the Radiation type as Isotropic in this lab. We can also define as Cosine Antenna, Pattern File, Pattern Data.
- If we chose Pattern File we can import pattern from EMPro, HFSS and CST as shown below (***we will do this in later lab***)



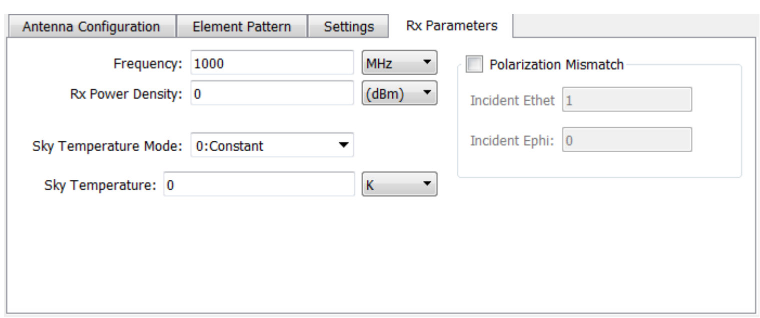
- Etha and EPhi are the polarization of antenna.

3. Under Settings define the parameters as below:



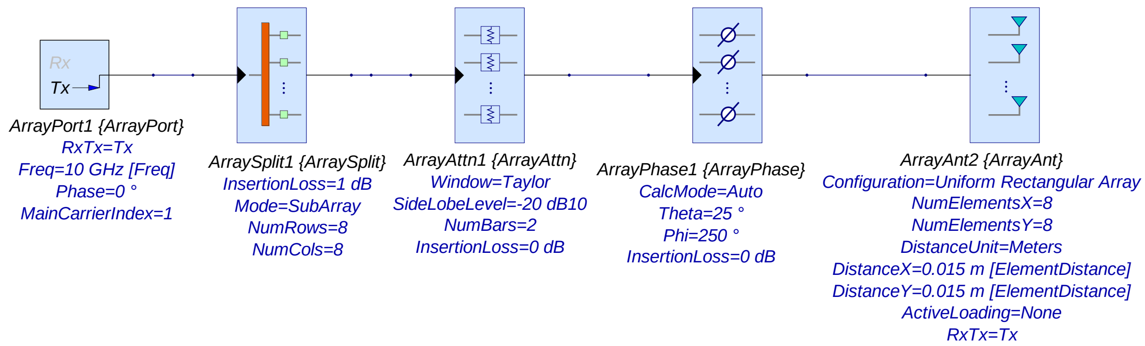
- Under Active loading we have defined none. However, if want to define antenna coupling matrix we can define under this tab. We can use a Coupling Matrix, S-Parameter file etc.
- Under Array Radiation we can define whether to have Upper Hemisphere or Full Sphere Radiation Pattern.

4. We are setting up Tx simulation so leave parameters under Rx Parameters as default as shown below

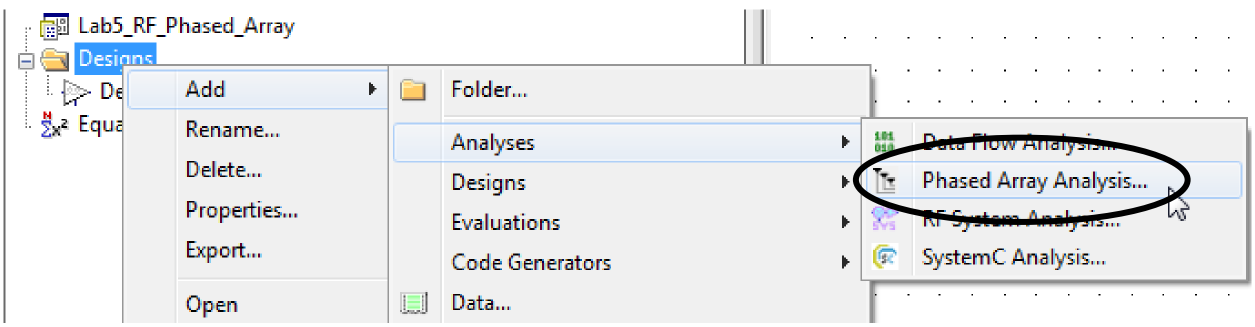


- Polarization Mismatch can be used if we want to analyze the loses in Receiver because of Polarization Mismatch.

Complete Phased Array Tx system should appear as shown below…..

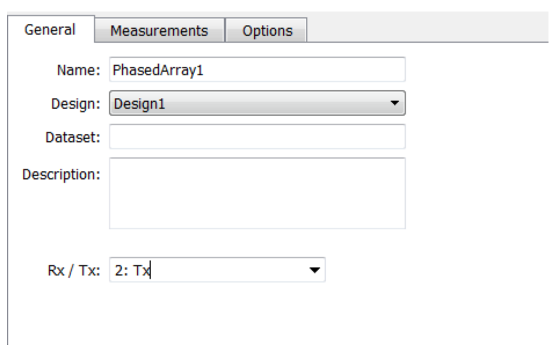


Step8: In the workspace tree **delete** Data Flow Analysis and add Phased Array Analysis as shown below:

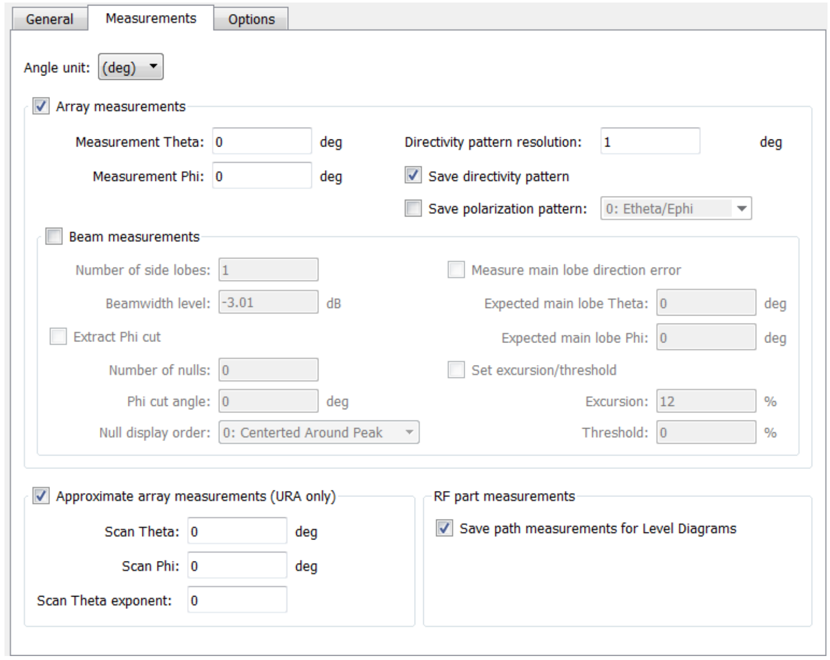


Step9: Double click on the Phased Array Analysis and define the parameters.

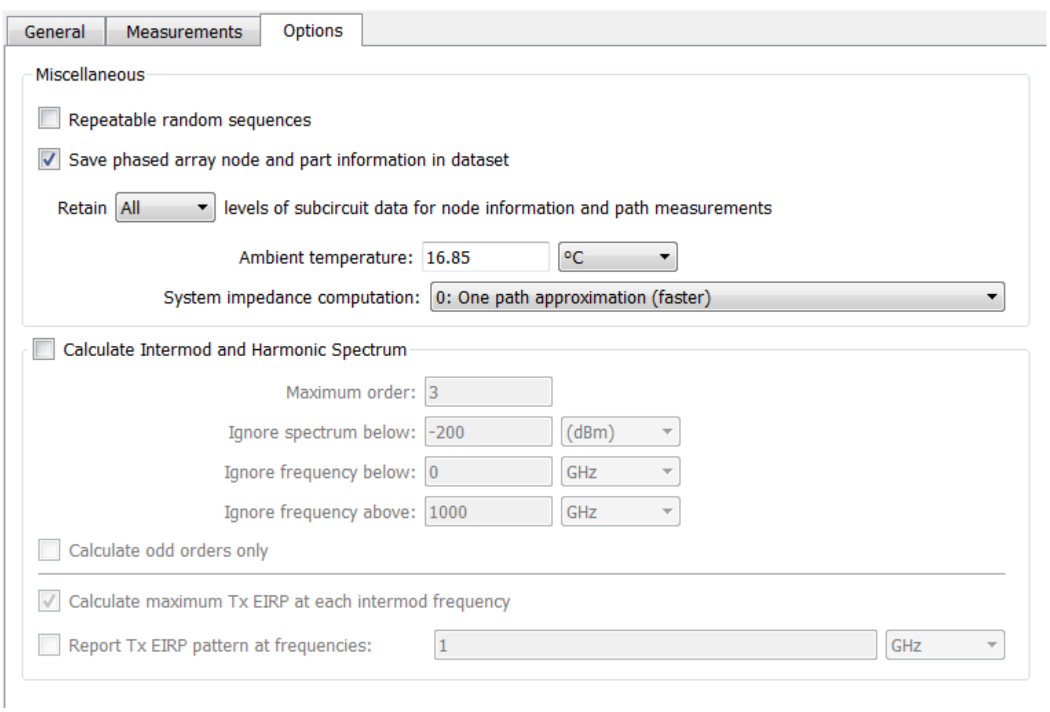
1. Under General Tab we will define whether it's a Rx or Tx chain.



2. Under Measurement tab define the following parameters:

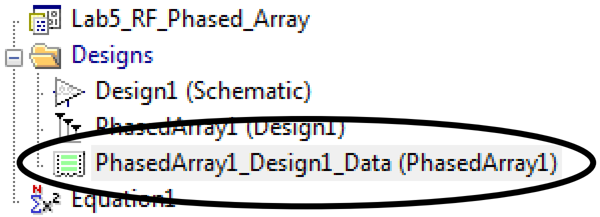


3. Under Option tab:

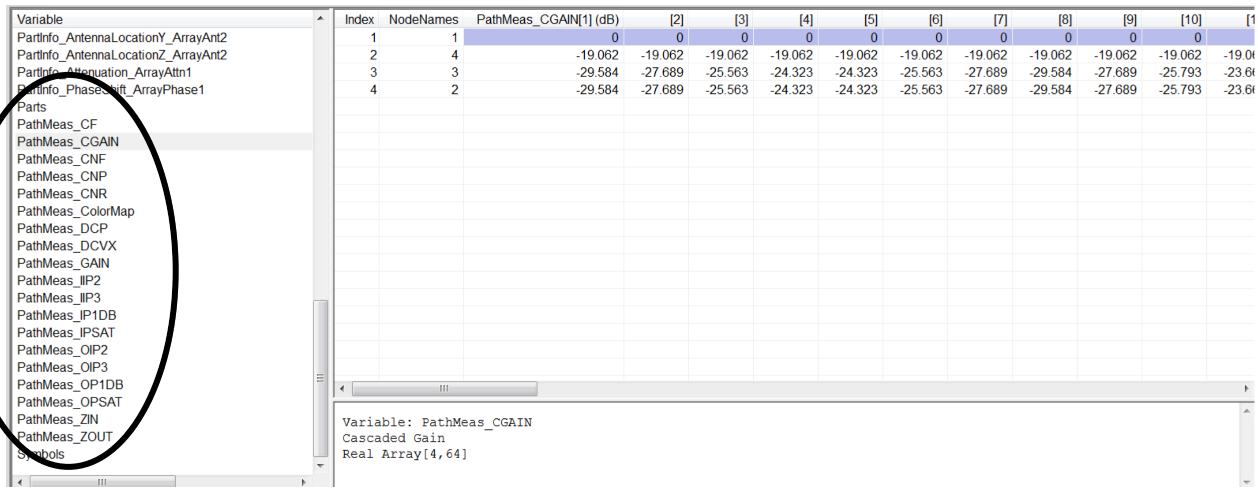


Click Apply and OK after finishing above setting.

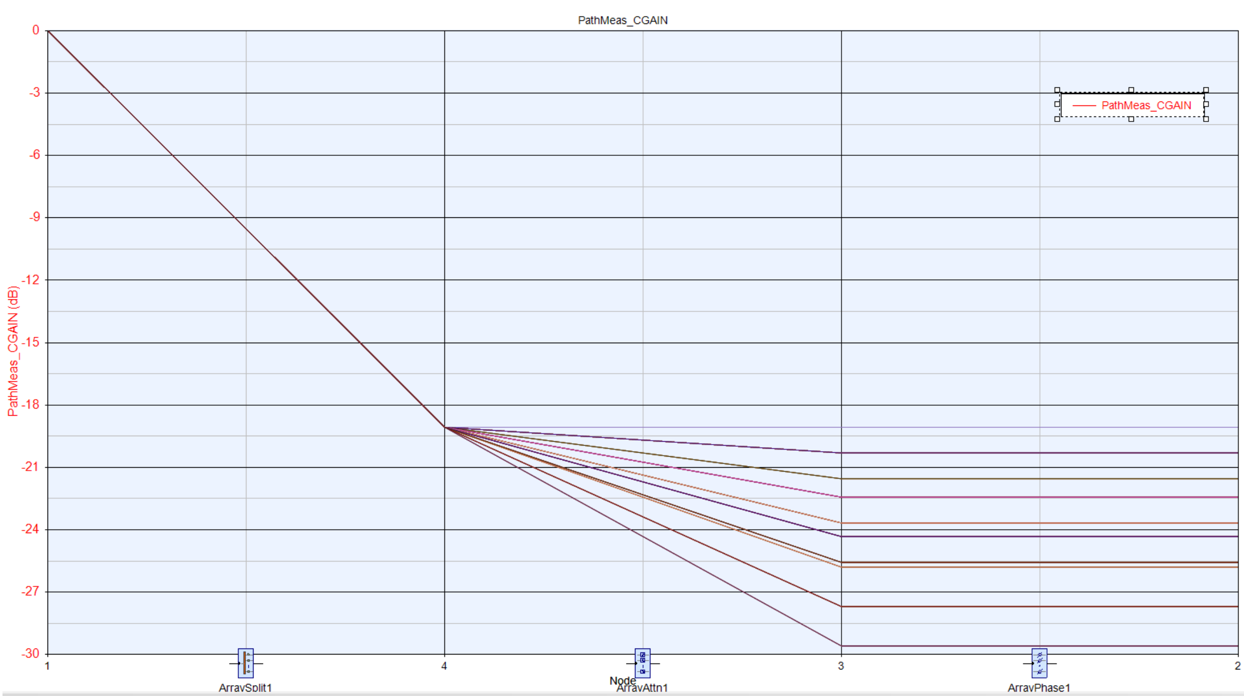
Step10: Run the Phased Array1 analysis. Once the simulation is finished a new dataset PhasedArray1_Design1_Data will be added.



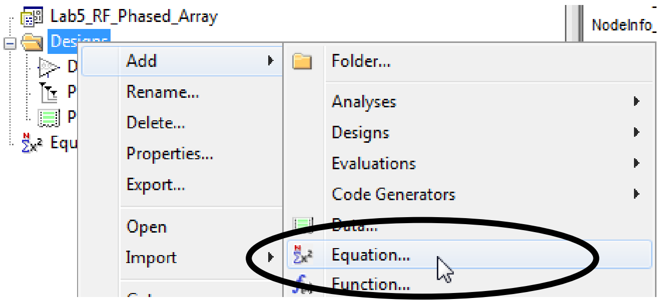
Step11: Open the dataset PhasedArray1_Design1_Data and we can see individual path measurement are calculated as shown below:

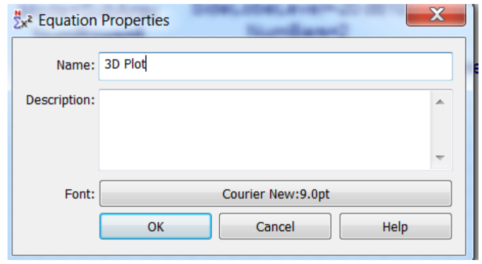


Step12: Right click on PathMeas_CGAIN→Add to Graph→New Graph.



Step13: To plot the 3D Far Field Pattern go to workspace tree, right click and add another Equations and name it as **3DPlot**

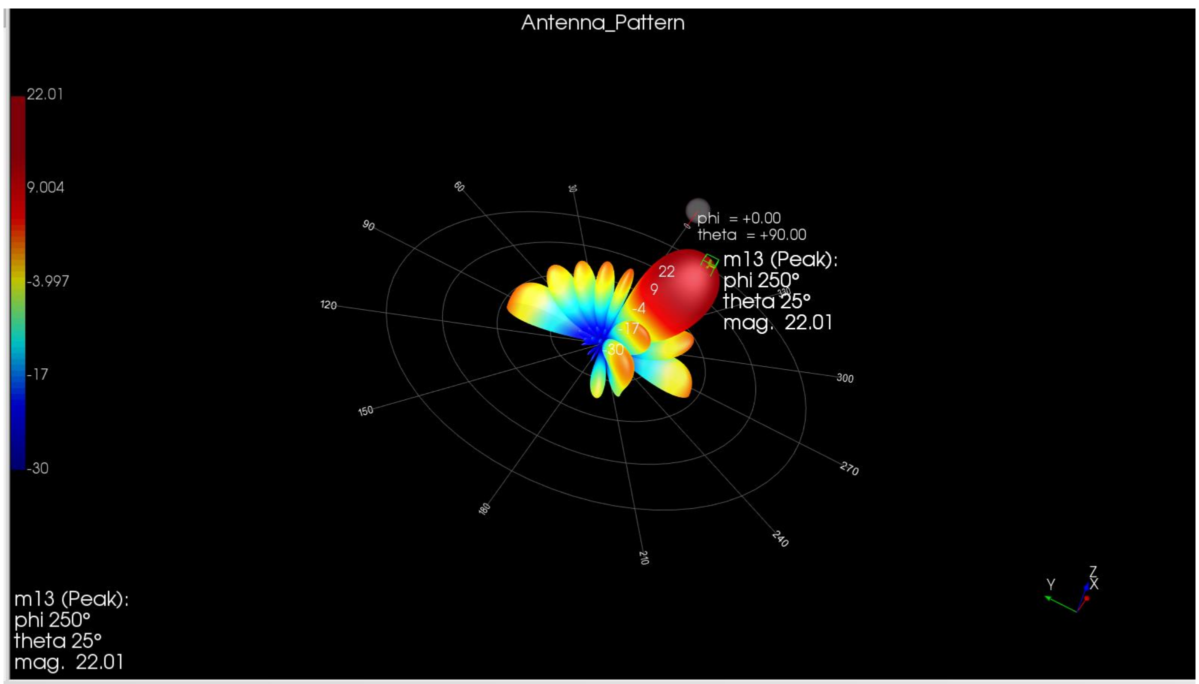




Step14: Double Click on 3D Plot (Equation) and write the below equations:

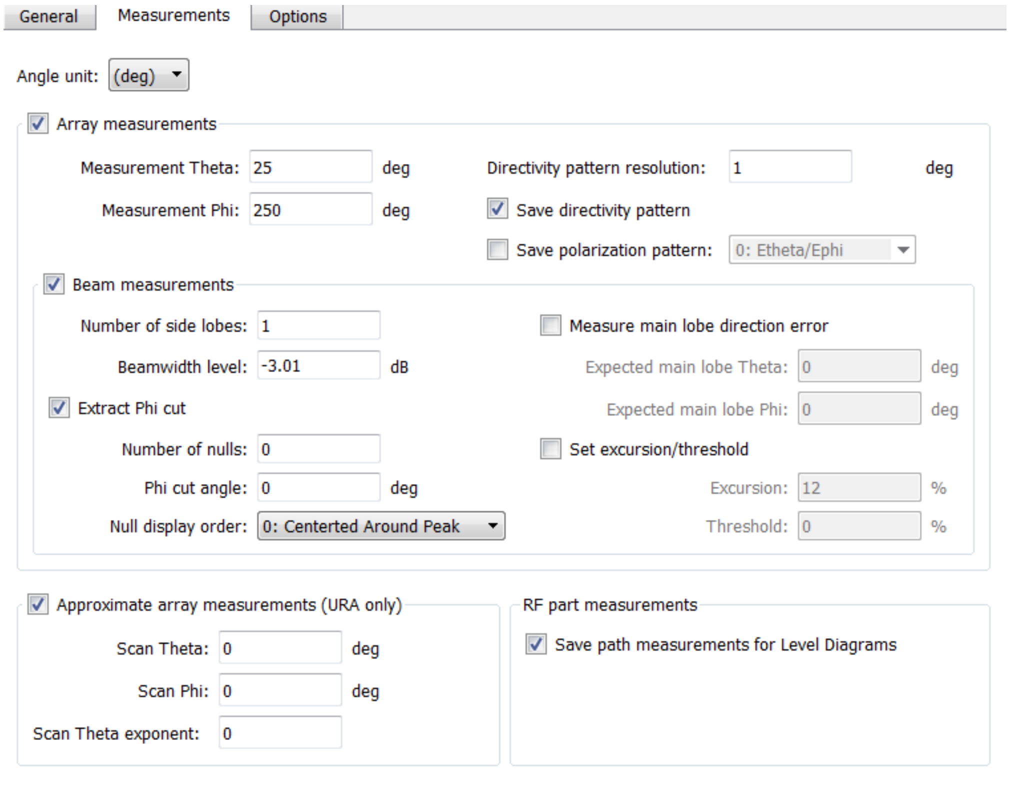
```
using('PhasedArray1_Design1_Data');

graph1=graph_figure('Antenna_Pattern');

graph1.add_spherical_data( ArrayMeas_Directivity_PhiGrid , ArrayMeas_Directivity_ThetaGrid,
max(10*log10(ArrayMeas_Directivity_Pattern),-30) );
```

Step15: Right Click on 3D Plot (Equations) under Workspace Tree and click on Run Equations. This will Plot the 3D Far Field Pattern as shown below:

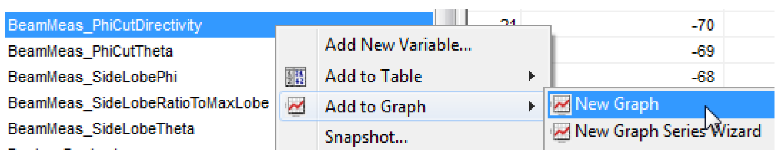


Right Click on the Main Beam and place a Peak Marker.

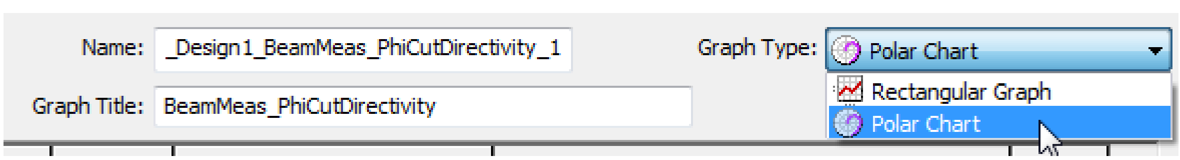
Step16: Now we will Plot the 2D Phi Cut in Rectangular and Polar Format. Double Click on Phased Array Analysis under workspace tree and under the Measurement Tab Define the below parameters. We have enabled the Extract Phi cut data. You can also enable main lobe direction error, if required.



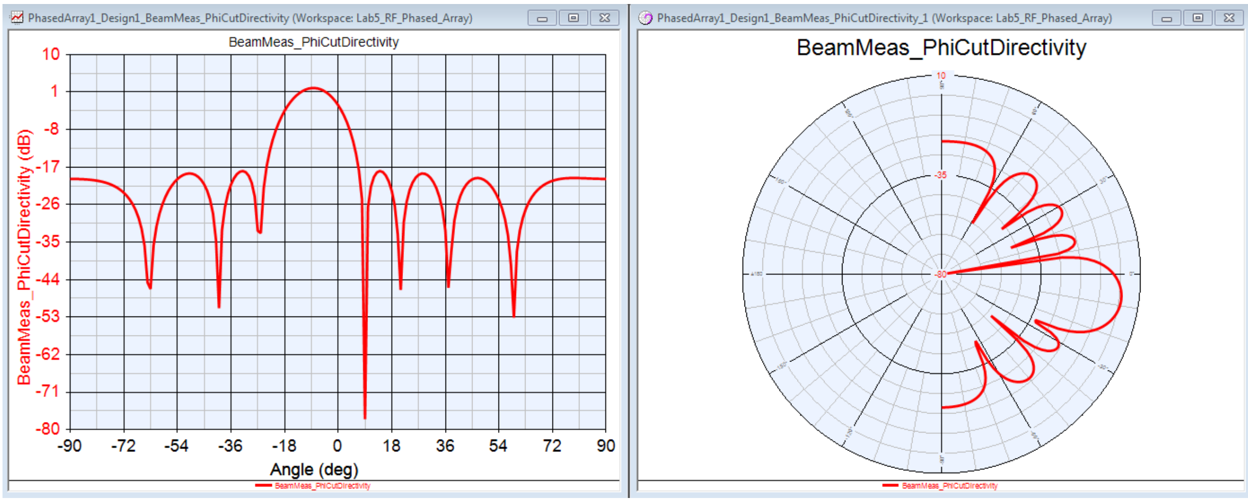
Step17: Run the analysis again and open the dataset PhasedArray1_Design1_Data. Right Click on **BeamMeas_PhiCutDirectivity**->Add to Graph->New Graph and plot the Rectangular plot



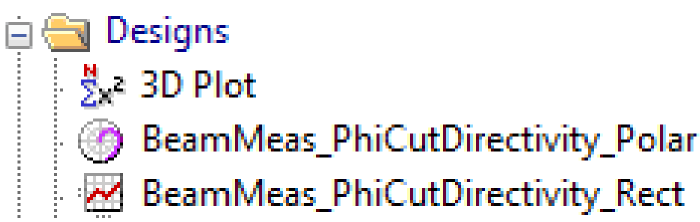
Repeat the same to add one more plot, double click on graph and change the plot type as Polar Chart and Polar graphs (select dB10 as Upper Radial Axis) as shown below:



Once done, we shall have following graphs in our workspace tree….



Step18: Rename these two graphs to something we can understand easily as shown below:



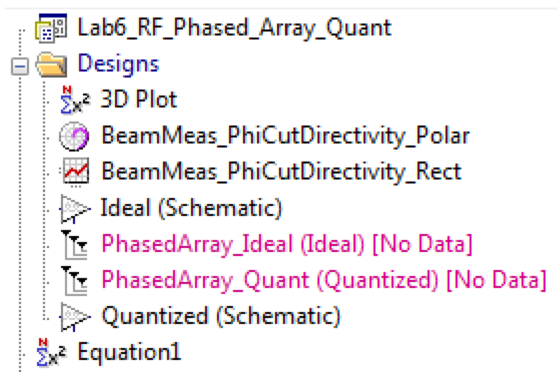
Lab 6: Quantization effects on Phased Array System

The Objective of this lab is to analyze phase shifters quantization effects on RF system performance and the Antenna Beam pattern.

Step1: Use **File->SaveAs** to save Lab5_RF_Phased_Array as **Lab6_RF_Phased_Array_Quant**

Step2: In Workspace Tree perform the following steps:

- Rename Design1 schematic as **Ideal**.
- Right click and copy/paste **Ideal** schematic and rename it as Quantized
- **Rename** PhasedArray1 Analysis as PhasedArray_Ideal. Double click on Phased Array_Ideal analysis and choose Design as "Ideal".
- **Copy/Paste** PhasedArray_Ideal analysis and rename it as PhasedArray_Quant. Double click on Phased Array_Quant analysis and choose Design as "Quantized".
- Delete the existing 3D plot showing Antenna Beam Pattern & dataset produced earlier to avoid any confusion.



Step3: Double click on Quantized (Schematic) to open the design. Double click on Attenuator & Phase Shifter models and define the Quantization and Number of Bits as below and click OK.

Name	Value	Units
CalcMode	0:Auto	
Theta	25	deg
Phi	250	deg
Quantization	1:Number of Bits (Uniform)	
NumBits	6	

Name	Value	Units
Window	10:Taylor	
SideLobeLevel	-20	dB
NumBars	2	
Quantization	1:Number of Bits (Uniform)	
NumBits	6	
StepSize	0.5	dB

Step4: Open 3D Plot (Equations) and modify the dataset name as **PhasedArray_Ideal_Ideal_Data** and graph title as **Antenna_Pattern_Ideal** as shown below:

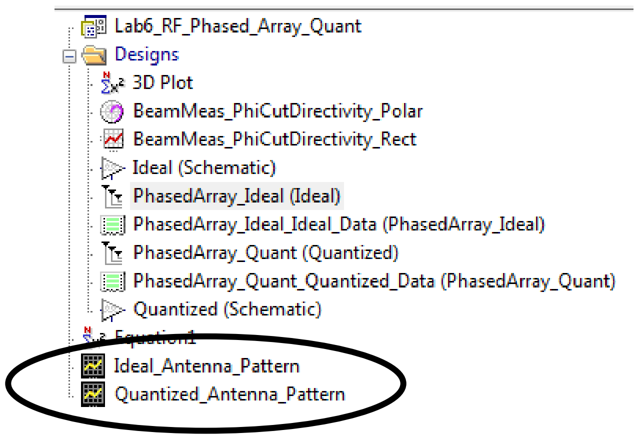
```
using('PhasedArray_Ideal_Ideal_Data');

graph1=graph_figure('Antenna_Pattern_Ideal');

graph1.add_spherical_data( ArrayMeas_Directivity_PhiGrid , ArrayMeas_Directivity_ThetaGrid,
max(10*log10(ArrayMeas_Directivity_Pattern),-30) );
```

Step5: Add below lines in the 3D Plot equations to enable 3D plot for Quantized results:

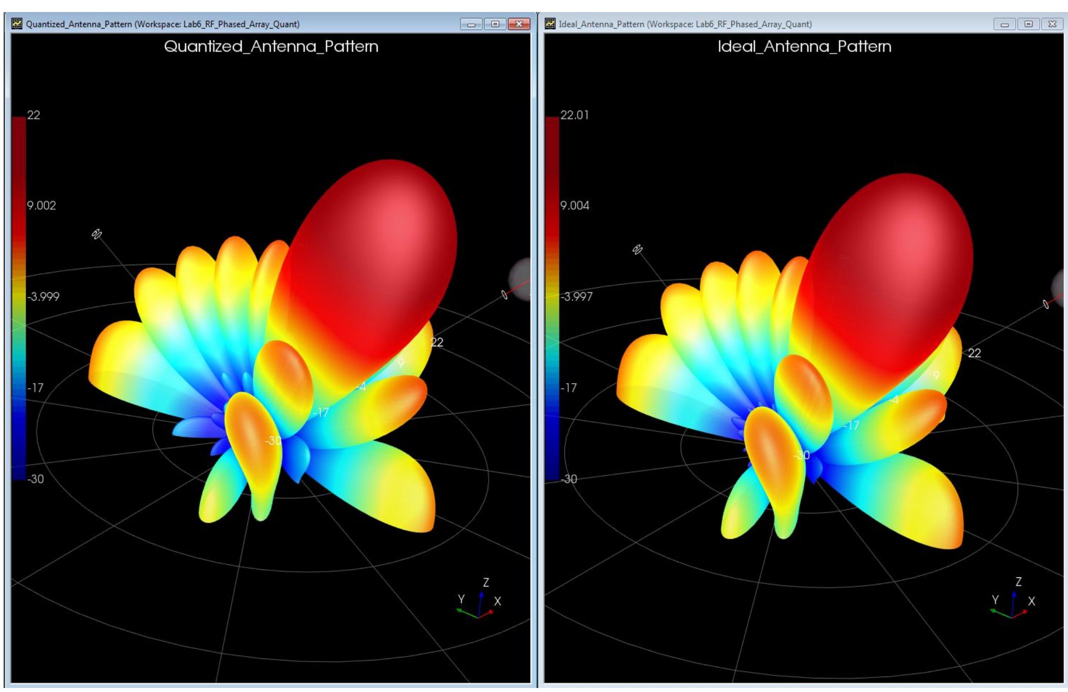
```
using('PhasedArray_Quant_Quantization_Data');

graph1=graph_figure('Quantized_Antenna_Pattern');

graph1.add_spherical_data( ArrayMeas_Directivity_PhiGrid , ArrayMeas_Directivity_ThetaGrid,
max(10*log10(ArrayMeas_Directivity_Pattern),-30) );
```

Step6: Run both PhasedArray_Ideal and PhasedArray_Quant analysis.

Step7: Once the simulation is finished, right click on 3D Plot and select Run Equations. This will add two new 3D Plots in the workspace tree as shown below:

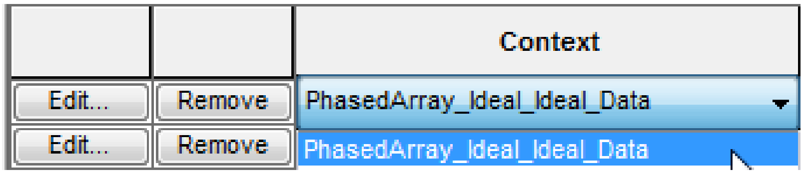


Step8: Open both the Far Field pattern side by side and compare. We can notice that the Side Lobes level seems to be different.

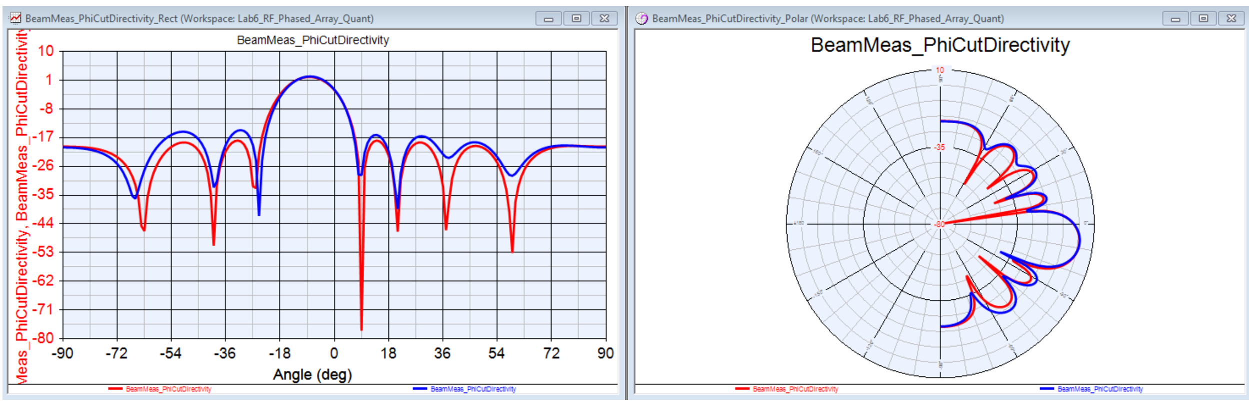


Step9: To get better comparison and see the effect of quantization, add BeamMeas_PhiCutDirectivity on the rectangular and polar plot.

***Note:* Due to dataset name change plots will not show any traces**…. double click on graph and change the Context to select proper dataset name as PhasedArray_Ideal_Ideal_Data as shown here…



Add new trace and select **BeamMeas_PhiCutDirectivity** from **PhasedArray_Quant_Quantized_Data dataset.** Repeat the same for Polar plot as we can clearly see the difference between Ideal and Quantization cases…..



Lab 7: Phase Shifters and Attenuator S-Parameter (S2P) effect on Phased Array System

Objective of this lab is to analyze the effects of vendor supplied S-Parameter measurement data of Phase Shifters and Attenuators on our RF system and eventually on the Antenna Beam. These components are frequency dependent and also provide different characteristics at different states hence by incorporating measured S-Parameter files for all the states we can model our RF system with higher degree of fidelity & accuracy. **Design tools usually will limit components to only have 1 S-Parameter file but with Digital Attenuator and Phase Shifter this is not the case as we will have 2^n S2P files** where “n” is the Quantization level or Number of Bits. Keysight SystemVue makes this job very easy for RF designers whereby we can use multiple S2P files representing state of the component as per the desired Beam Steering direction.

Step1: Use File->SaveAs to save Lab6_RF_Phased_Array_Quant as Lab7_RF_Phased_Array_SParam

Step2: Double click on Attenuator and change the NumBits = 5. **Similarly,** modify the Phase Shifter component to have NumBits = 4.

Note: We are doing this purely due to the S-Parameter data we are going to use for our workshop. Look under the workshop folder provided on your PCs to find S-Parameter files for below parts:

<u>HMC1019ALP4E: 5-bit MMIC Digital Attenuator (Analog Devices):</u>

www.analog.com/media/en/technical-documentation/data-sheets/hmc1019a.pdf

<u>HMC543ALC4B: 4-bit MMIC Phase Shifter (Analog Devices):</u>

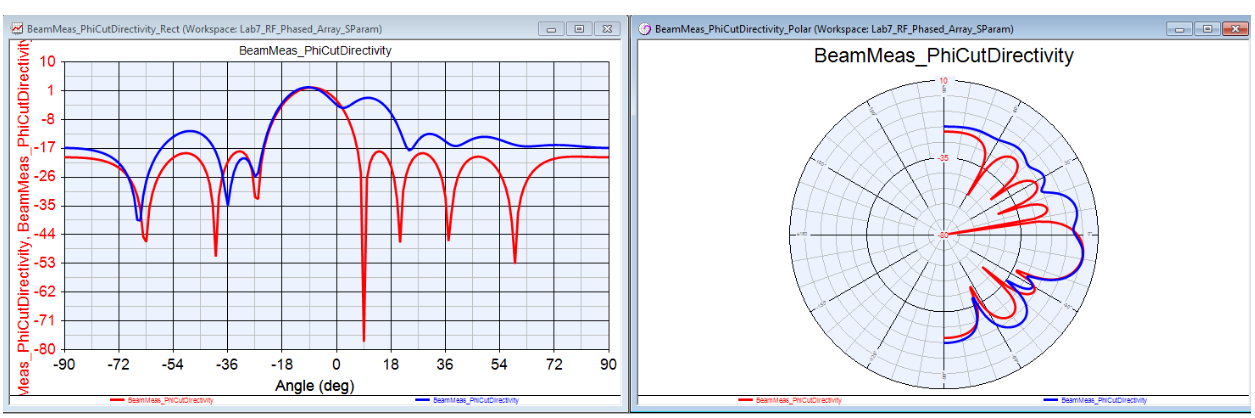
www.analog.com/media/en/technical-documentation/data-sheets/HMC543A.pdf

These files in zip format can also be found under SVue examples folder:

C:\Program Files\Keysight\SystemVue2017\Examples\RF Architecture Design\Phased Array Analysis

---

Step3: Run Simulation and observe the changes due to these changes:



Let's do some changes to our designs so that it is easier to change beam direction and run analysis to observe the corresponding results.

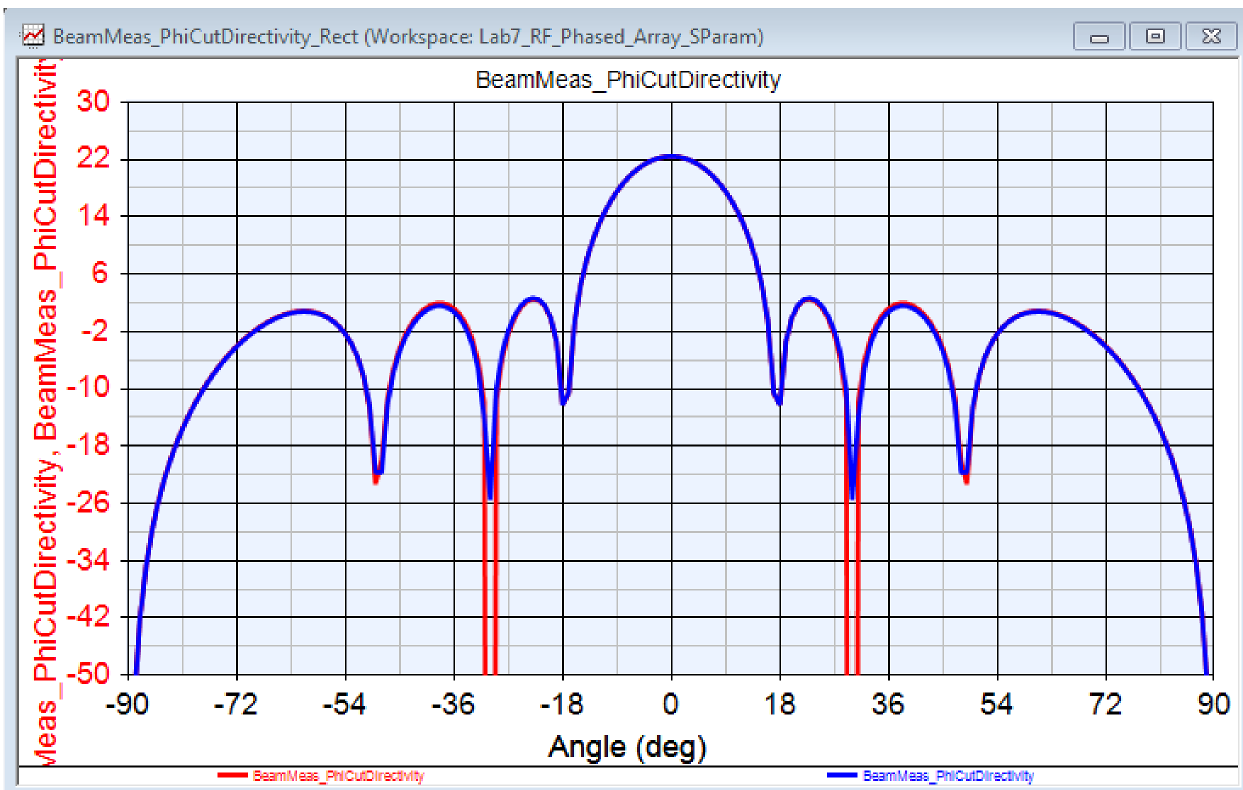
a. Right click on 3D Plot (equation page) and **uncheck** "Auto-Calculate"
b. Right click on Equations1 and **uncheck** "Auto-Calculate", double click and add following lines to existing variables:

```
Theta=0;
Phi=0;
runanalysis('PhasedArray_Ideal');
runanalysis('PhasedArray_Quant');
```

c. Open Ideal and Quantized design schematics and define variables Theta and Phi instead of 25deg and 250deg in the Phase Shifter component of both the designs

Step4: Open Equations1 page (if it is not open already) and run script by either pressing CTRL+G ***or*** right click and select Run Equations

Observe the plots and we can notice that results match reasonable well as shown below with difference observed in the NULL levels as we expect.



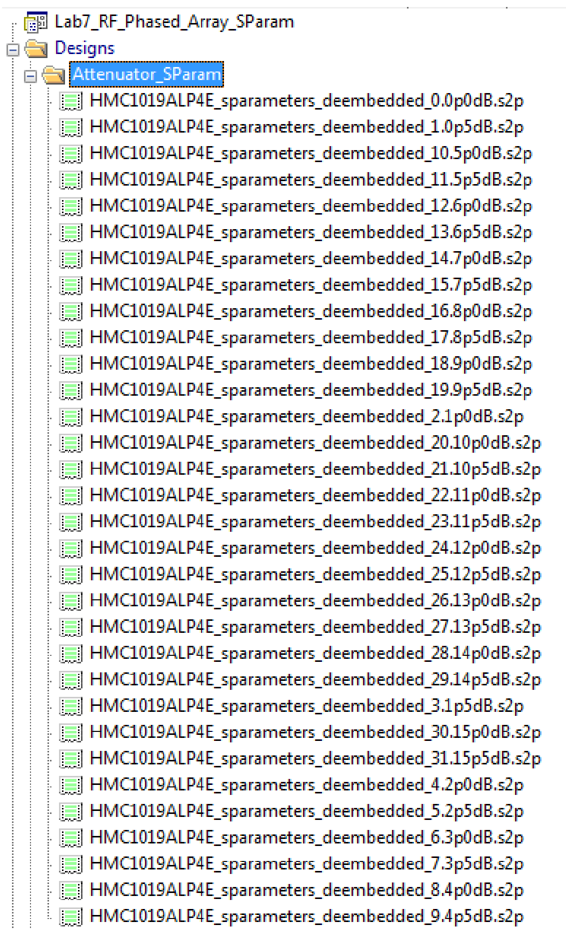
Try changing Theta & Phi values in Equations1 page and notice how well Ideal and Quantized results match at different combinations. You will notice that results match well for some cases and may not match well for other cases, **think WHY?**

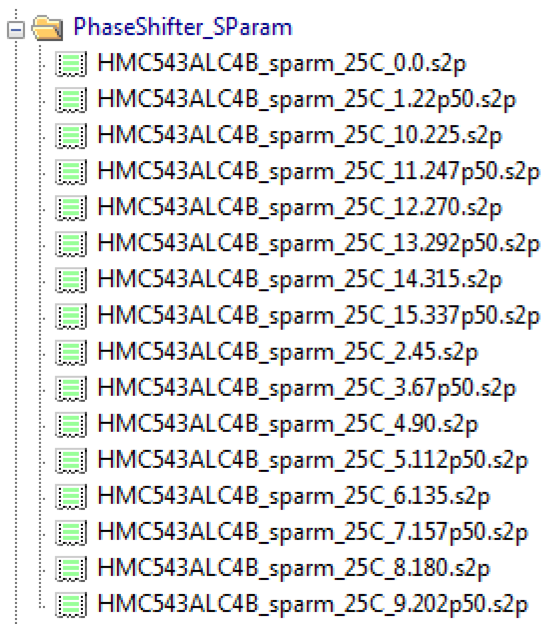
Step5: As we have observed the effects of the Quantization onto the beam pattern on various Theta & Phi it is time to include realistic S-Parameter data of these 2 critical components of phased array system and observe its impact on our system performance.

Right click on Designs folder under workspace tree and select Add->Folder to add 2 folders with name **Attenuator_SParam** and **PhaseShifter_SParam**.

Right click on **Attenuator_SParam->Import->S-Data File….** and browse to **<u>HMC1019ALP4E</u>** folder in the provided workspace and select all the S2P files by pressing SHIFT key and click Open. This step will import all the data files locally to our workspace as shown here, note there are 32 S2P files representing 32 states of attenuator with 0.5dB steps. Repeat the same for Phase Shifter by going to folder **<u>HMC543ALC4B</u>** and import all 16 S2P files representing 16 phase states of digital phase shifter.

Once done our workspace tree will have following files…..

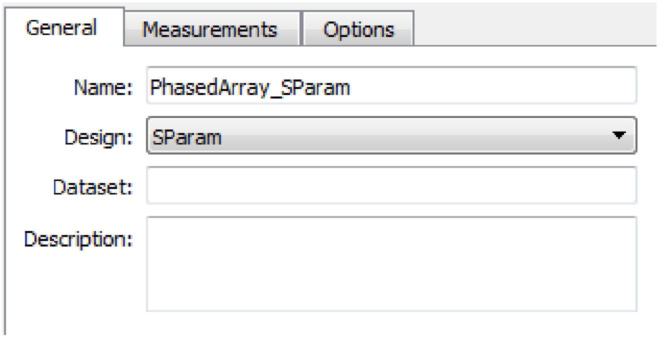




Note: It is not necessary to import the S2P files inside the workspace and we can work with external files approach as well.

Step6: Copy/Paste Quantized schematic design and its corresponding PhasedArray simulator in the workspace tree. **Rename schematic as SParam and its simulator as PhasedArray_SParam**.

Step7: Double click on PhasedArray_SParam simulation controller and select the design as **SParam**



Step8: Open SParam schematic, double click on Attenuator component and set parameter as shown here.

- Select UseSParameters = YES
- SParamSource = Dataset (if using external S2P files without importing the same in workspace select source as File)
- Define the Dataset name as 1st file in the list representing 0dB state: **HMC1019ALP4E_sparameters_deembedded_0.0p0dB.s2p**

Name	Value	Units
Window	10:Taylor	
SideLobeLevel	-20	dB
NumBars	2	
Quantization	1:Number of Bits (Uniform)	
NumBits	5	
StepSize	0.5	dB
UseSParameters	1:YES	
SParamSource	1:DataSet	
DatasetName	HMC1019ALP4E_sparameters_deembedded_0.0p0dB.s2p	

Step9: Double click on the Phase Shifter component and do setting as shown below. Provide Dataset name as 1st file in the list representing 0deg state: **HMC543ALC4B_sparm_25C_0.0.s2p**

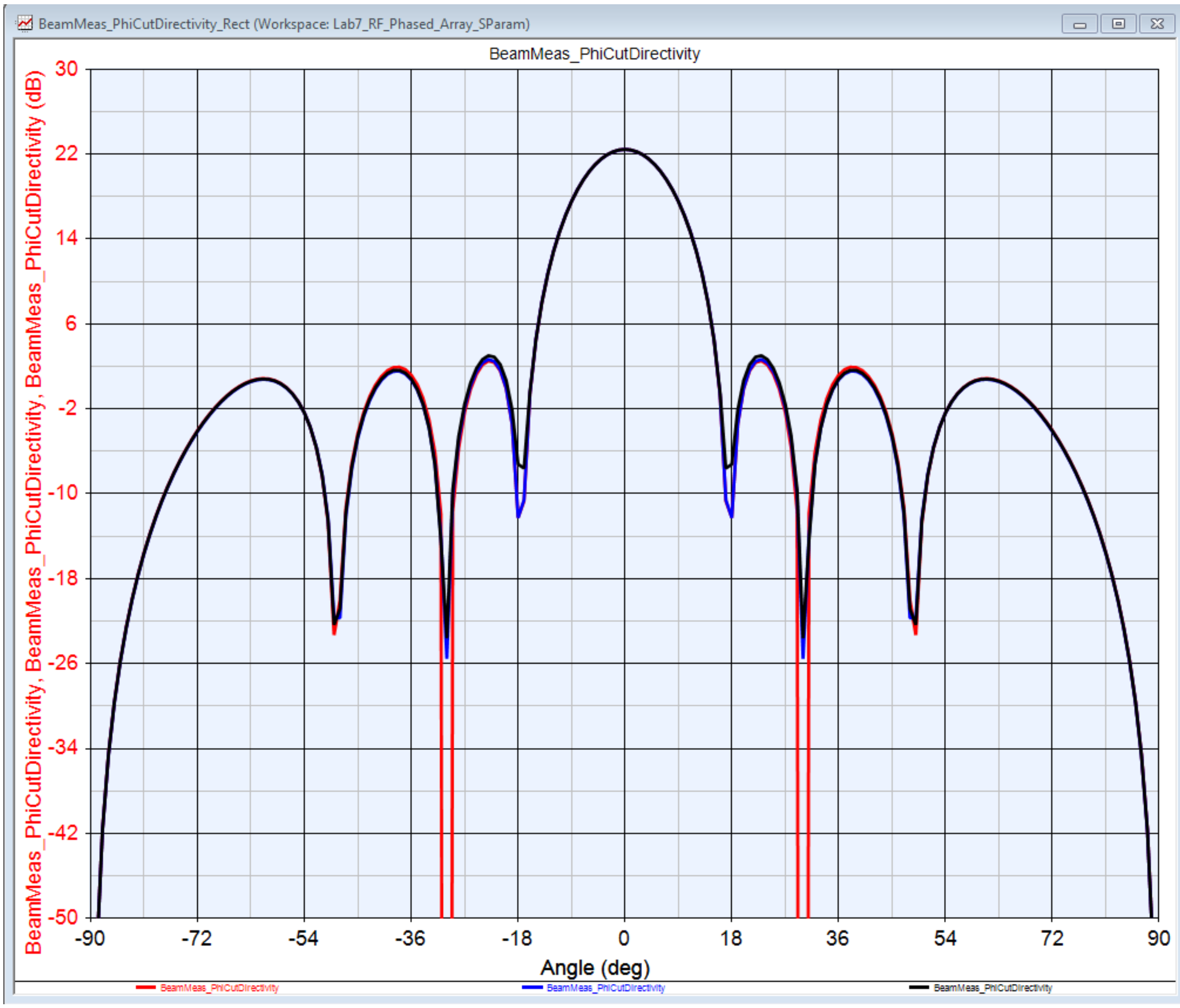
Name	Value	Units
CalcMode	0:Auto	
Theta	Theta	deg
Phi	Phi	deg
Quantization	1:Number of Bits (Uniform)	
NumBits	4	
UseSParameters	1:YES	
SParamSource	1:DataSet	
DatasetName	HMC543ALC4B_sparm_25C_0.0.s2p	

Step10: Run Simulation and notice new dataset with name “PhasedArray_SParam_SParam_Data” generated after successful simulation.

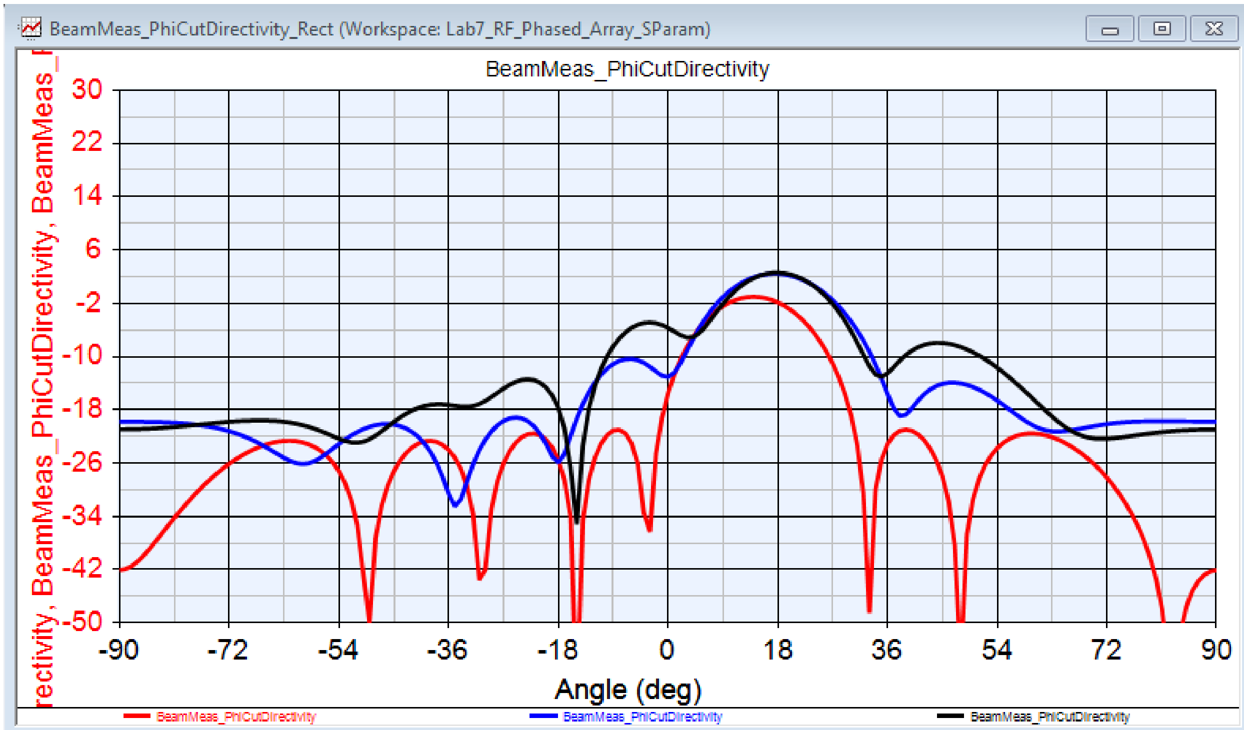
Step11: Double click on 3D Plot equation page and add following lines to the existing code:

```
using('PhasedArray_SParam_SParam_Data');

graph1=graph_figure('SParam_Antenna_Pattern');

graph1.add_spherical_data( ArrayMeas_Directivity_PhiGrid , ArrayMeas_Directivity_ThetaGrid, max(10*log10(ArrayMeas_Directivity_Pattern),-30) );
```

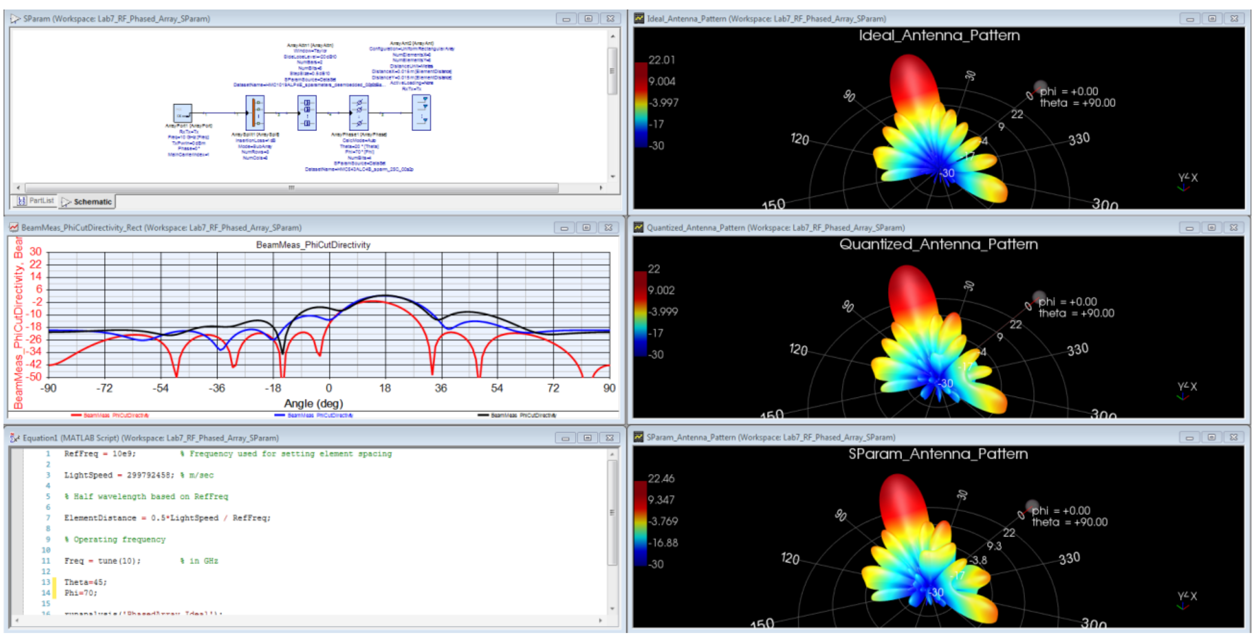
Step12: Open PhasedArray_SParam_SParam_Data dataset, select BeamMeas_PhiCutDirectivity, right click and add it on BeamMeas_PhiCutDirectivity_Rect (or whatever name you provided earlier) graph.



Step13: Open Equation1 page, add one more command as runanalysis('PhasedArray_SParam'). Define Theta=45, Phi=70 and run equations again to observe the Phi Cut plot……



Step14: Try other Theta & Phi combinations and run equations to observe the response. When done, right click on 3D Plot and select Run Equations. Open all 3D plots from workspace tree and select Window->Tile Horizontal to observe all windows together



Step15: Open all 3 datasets and compare other measurements such EIRP etc…We can also right click and select option of Add to Table and send other 2 dataset results on the same table for quick and better comparison.

Step16: Set Theta = 0 and Phi = 0 under Equation1 page and run the Equations again.

Note: S-Parameter File name convention while using multiple S2P files for Attenuator & Phase Shifter components

1. When applying S-parameter s2p files to describe the quantized states, make sure the number of indexed s2p files are the same as the number of quantized states.
2. Indexed s2p files must follow the naming convention. For example, S-parameters for 4-bit attenuator can be organized as:

    Attenuator4bit_0.0p0dB.s2p
    Attenuator4bit_1.0p5dB.s2p
    Attenuator4bit_2.1p0dB.s2p
    Attenuator4bit_3.1p5dB.s2p
    Attenuator4bit_4.2p0dB.s2p
    Attenuator4bit_5.2p5dB.s2p
    Attenuator4bit_6.3p0dB.s2p
    Attenuator4bit_7.3p5dB.s2p
    Attenuator4bit_8.4p0dB.s2p
    Attenuator4bit_9.4p5dB.s2p
    Attenuator4bit_10.5p0dB.s2p
    Attenuator4bit_11.5p5dB.s2p
    Attenuator4bit_12.6p0dB.s2p
    Attenuator4bit_13.6p5dB.s2p
    Attenuator4bit_14.7p0dB.s2p
    Attenuator4bit_15.7p5dB.s2p

The **trailing integer value** in the sub-string before the leftmost dot is treated as the unique index identifier of the file. **You can provide detailed information in between the first dot and the last file extension.**

If S-parameters is per state (for example, 4-bit phase shifter has 16 states), the S-parameters files (or datasets) need to be named based on the naming convention:

FileName#.Info.moreInfo.sNp,

where "#" represents state index and "N" represents S-parameters port number.

The **FileName** or **DatasetName** parameter only needs to specify **the first S-parameters** file (or dataset) among the available states.

3. Do **NOT** use Attenuator4bit**1.5dB**_0.s2p, because "1" after "bit" and before ".5dB" will be treated as index.
4. When using S-parameter to describe the quantized attenuation values, the phase of the S-parameter will also be applied.
5. Based on the quantized attenuation state for each channel, the corresponding S-parameters file (dataset) will be automatically selected to model that channel in a phased array simulation.

S-parameters Related Parameters

- **SParamSource**
  SParamSource specifies the source of S-parameters. It can be either **File** or **Dataset**.
  - **FileName**
    Specifies S-parameters file name (relative to the same workspace directory) or file path when SParamSource = File.
  - **DatsetName**
    Specifies S-parameters dataset name. S-parameters can be imported into workspace dataset using **File > Import > S Data File ...**
- **InterpDomain**
  Interpolation domain:
  - **Data based** - defined by S-data file format.
  - **Polar** - S-data interpolation in polar coordinates.
  - **Polar DB** - S-data in dB interpolation in polar coordinates.
  - **Rectangle** - S-data interpolation in rectangle coordinates (Re(S), Im(S)).

- **InterpMode**
  Interpolation method:
  - **Linear** - using linear interpolation
  - **Spline** - using cubic spline interpolation
  - **Cubic** - using cubic polynomial interpolation
- **ExtrapMode**
  Extrapolation method:
  - **Constant** - using closest data point in all extrapolation area.
  - **Interpolation** - using interpolation domain method for extrapolation.

- **RxInOrTxOutMap**
  Specifies the S-parameters port number corresponding to Rx input when the model is used in Rx mode or the Tx output when the model is used in Tx mode.
- **RxOutOrTxInMap**
  Specifies the S-parameters port number corresponding to Rx output when the model is used in Rx mode or the Tx input when the model is used in Tx mode.

  Suppose **RxInOrTxOutMap = 2** and **RxOutOrTxInMap = 1.** Under Rx mode, it means S-parameters port 2 is the input port and S-parameters port 1 is the output port, in the signal propagation direction across the phase shifter. Under Tx mode, it means S-parameters port 2 is the output port and S-parameters port 1 is the input port, in the signal propagation direction across the phase shifter.

Lab 8: Amplifier Non-Linearity effect on Phased Array System

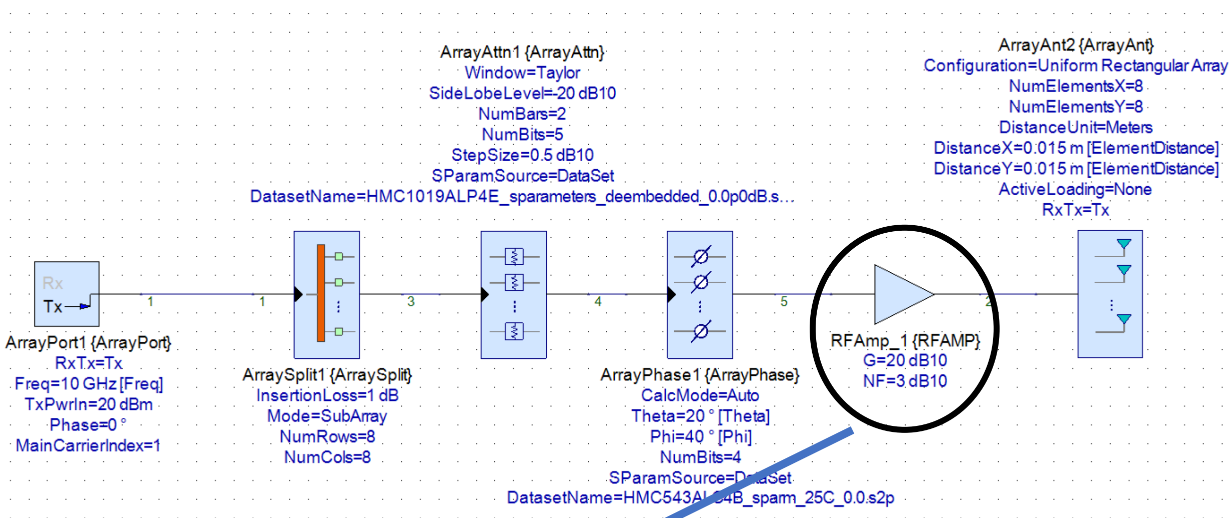
Objective of this lab is to analyze the effect of Amplifier Nonlinearities like 1-dB compression and AM-PM effect on the Antenna Beam.

Step1: Use **File->SaveAs** to save Lab7….. as **Lab8_RF_Phased_Array_AmpNL**

Step2: Copy/Paste SParam schematic and PhasedArray_SParam simulation controller. Rename copied schematic as "**SParam_and_Amp**" and simulation controller as "**PhasedArray_SParam_Amp**"

Step3: Double click **PhasedArray_SParam_Amp** simulation controller and select the design as **SParam_and_Amp**

Step4: Open SParam_and_Amp schematic. Insert an Amplifier (from RF Design library) between Array Phase Shifter and Array Antenna Model as shown below. Define the Amplifier parameters as below:



Name	Value	Units
G	20	dB
NF	3	dB
OP1dB	10	dBm
OPSAT	13	dBm
OIP3	20	dBm
OIP2	30	dBm
RISO	50	dB
EnablePN	0:NO	
AMtoPM_Mode	1:PointDerivative	
AMtoPM	-5	

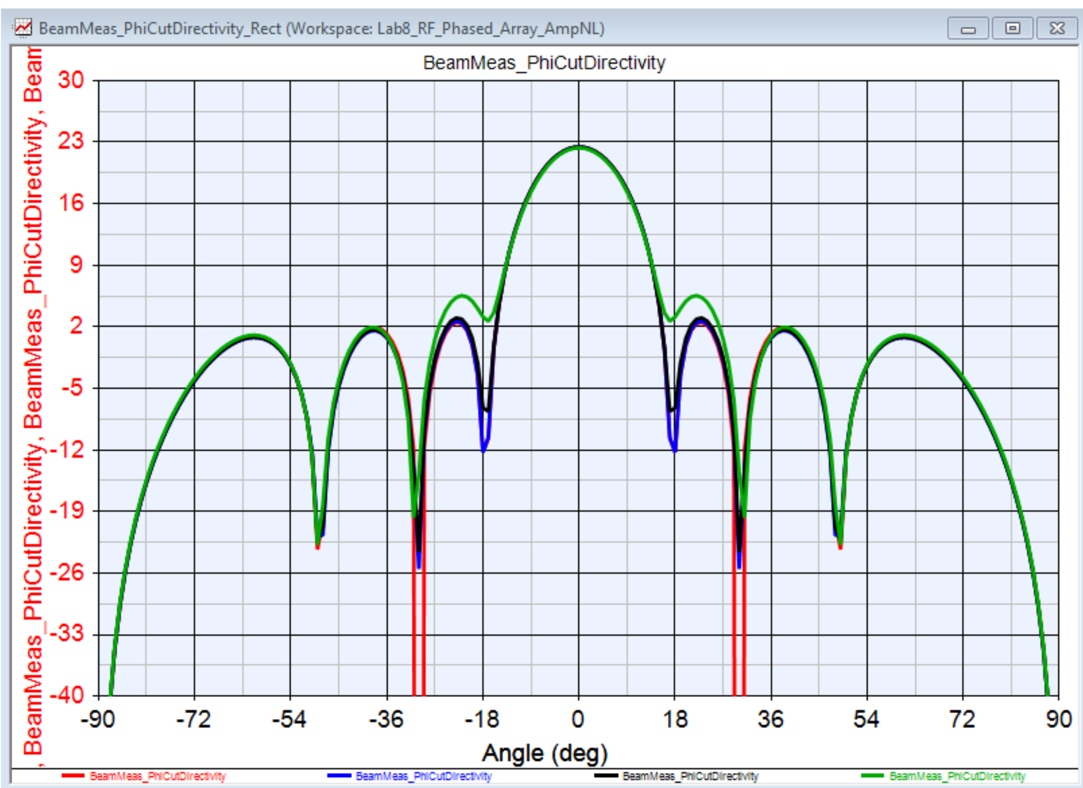
Step5: Double click on Tx source component and change the input power to 20 dBm

Name	Value	Units	Default	Ise Defaul	Tune	Show
RxTx	2:Tx		1:Rx	☐	☐	☑
Freq	Freq	GHz	[1000] MHz	☐	☐	☑
TxPwrIn	20	(dBm)	[0] dBm	☐	☐	☑
Phase	[0]	(deg)	[0] °	☑	☐	☑
MainCarrierIndex	1		1	☑	☐	☑
Z0	50	(Ohm)	50 Ω	☑	☐	☐
ZFreq	0	Hz	0 Hz	☑	☐	☐
PORT	1		1	☐	☐	☐
Ta		°C	°C	☑	☐	☐

Step6: Double Click on 3D Plot equation page and write the below equations.

```
using('PhasedArray_SParam_Amp_SParam_and_Amp_Data');

graph1=graph_figure('Amp_Antenna_Pattern');

graph1.add_spherical_data( ArrayMeas_Directivity_PhiGrid , ArrayMeas_Directivity_ThetaGrid,
max(10*log10(ArrayMeas_Directivity_Pattern),-30) );
```

Step7: Run Simulation. Open the **PhasedArray_SParam_Amp_SParam_and_Amp_Data** dataset and select BeamMeas_PhiCutDirectivity onto the 2D Phi cut rectangular plot we created earlier



Notice the side lobe level due to Amplifier's non-linearity.

Step8: Open Equation1 page and add one more analysis to be run when we change Theta and Phi values. Add command: **runanalysis('PhasedArray_SParam_Amp');** to the existing commands/variables.

Step9: Set other values of Theta & Phi and run the equations to observe results.

Notice how we started with a kind of ideal Phased Array RF system and turned 1 knob at a time to make it more realistic and practical. This process allows us to make some of very important System Architecture level engineering decisions reducing the troubleshooting time later in the system development cycle.

Lab 9: Phased Array Element Failure analysis using Monte Carlo

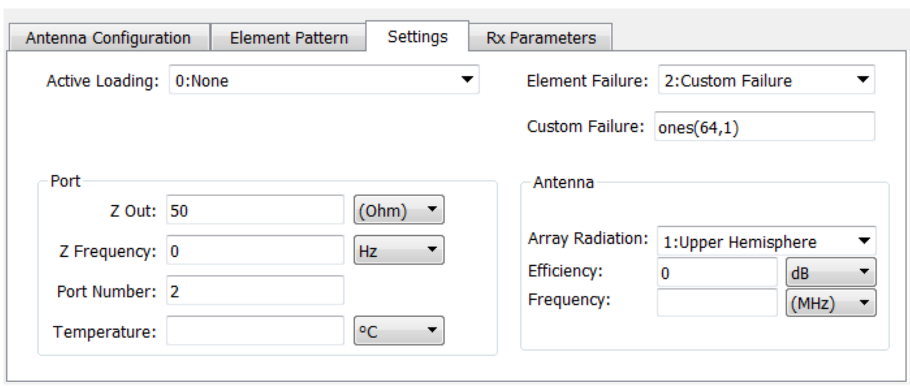
Phased Array system degrade graciously when some of the elements or the TR Modules fail due to some reason and Phased Array System continues to work even with those failed elements with reduced performance levels. Objective of this lab is to study the effect of Element Failure on a Phased Array System.

Step1: Open Lab5….. and save it as **Lab9_MonteCarlo.**

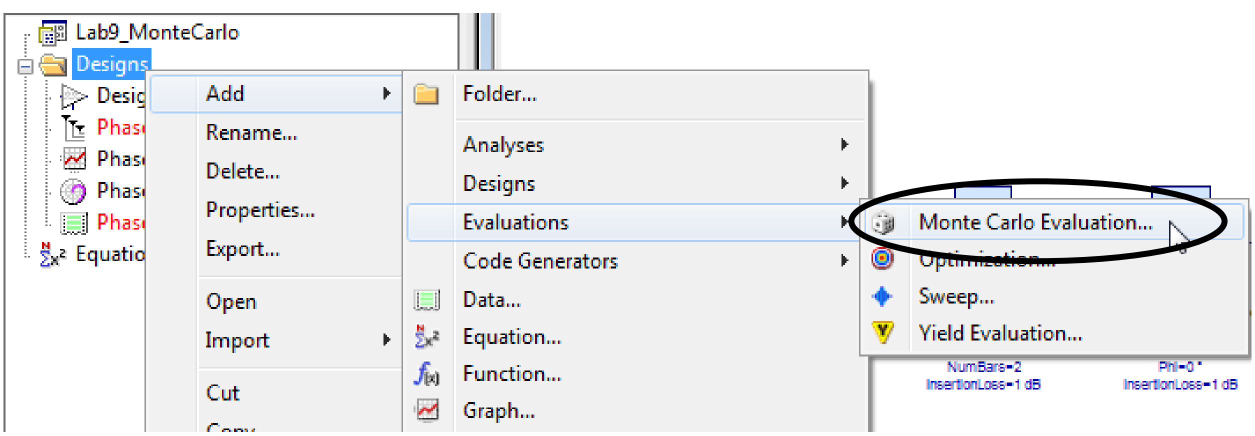
Note: We are using the near ideal RF system for this exercise so that we can clearly see the effects of element failure however if designers want they can continue from the last lab where we have achieved higher RF modeling fidelity by incorporating Quantization, S2P files for Digital Attenuator & Phase Shifter and accounted for Amplifier Non-Linearities.

Step2: Delete 3D Plot equations page and existing 3D plot

Step3: Open the Design1 (Schematic) and double click on Phased Array Antenna Model. Under Settings tab we will define the Element Failure type = Customer Failure and set Custom Failure = ones(64,1) as shown below:



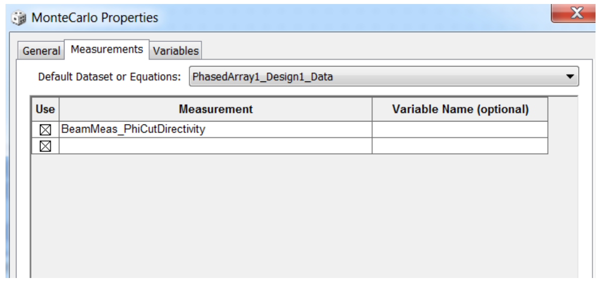
Step4: Right Click on Designs Folder under Workspace Tree and add Monte Carlo Evaluation.



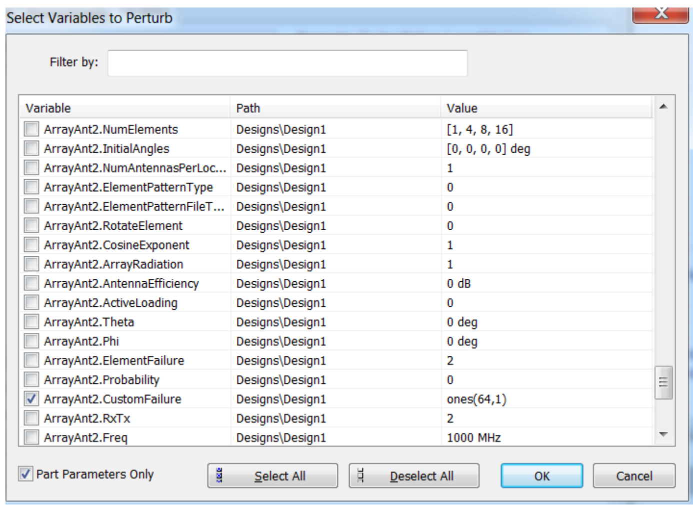
Step5: Under Workspace Tree, double click on MonteCarlo1 to change the parameters.

- Under General tab, select **PhasedArray1** analysis under Analysis to Calculate

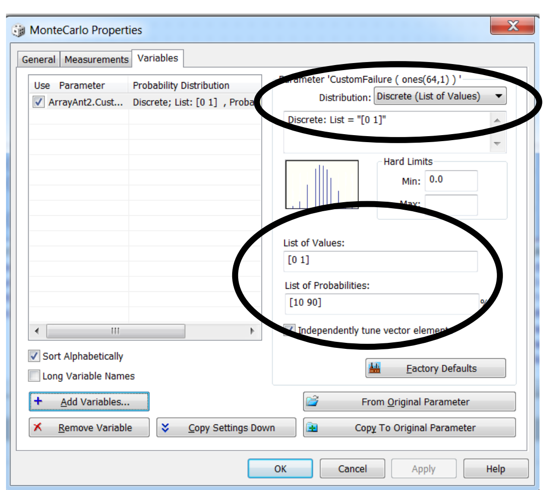
- Under Measurement tab, manually enter **BeamMeas_PhiCutDirectivity** under Measurement (***You can open dataset and refer to the desired measurement name on which we would like to see the impact of element failures***)



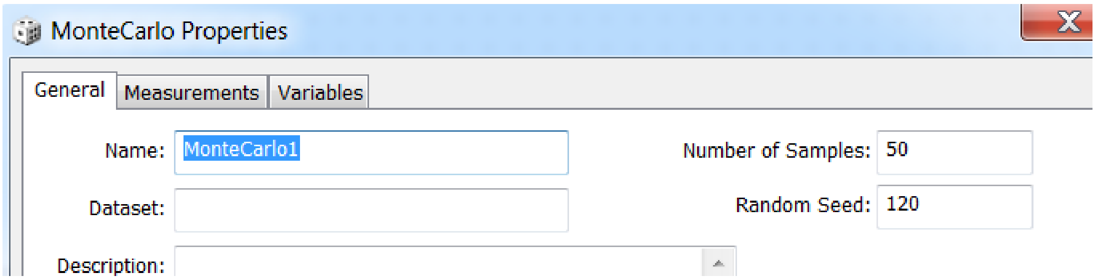
- Under Variables tab click on Add Variable→Select Variable.
- Select **ArrayAnt2.CustomFaliure** and click OK.



- Define **Distribution as Discrete** (List of Values)
- **Define List of values** and List of Probabilities as below. Here we are defining 10% failure rate, with the combination of list of values as [0 1] and probabilities of [10 90], it means that 10% of the elements will have values of 0 signifying failure.

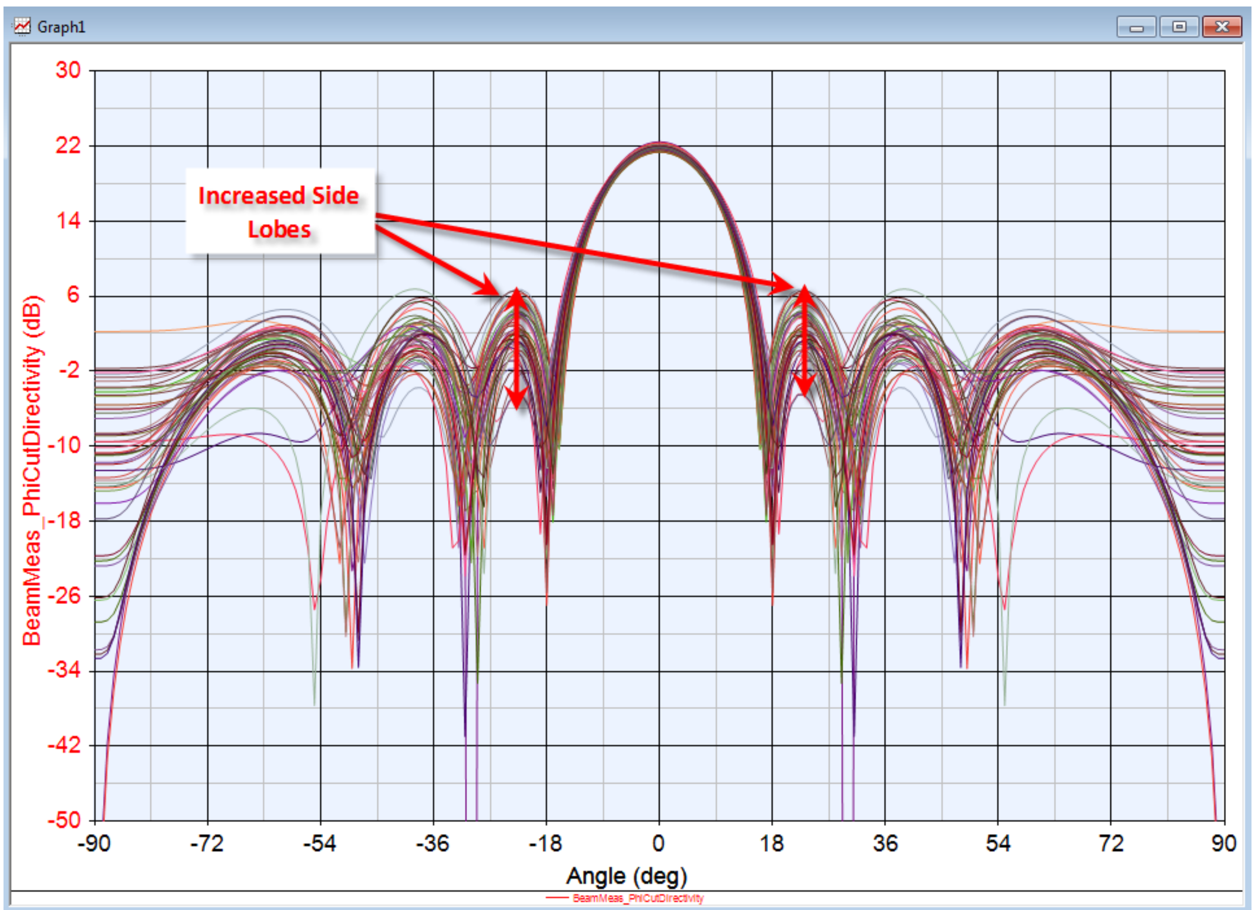


- Under General Tab, set Number of Samples = 50 and some Random Seed (e.g. 120)



Step6: Right Click on MonteCarlo1 under Workspace Tree and RUN it. It will take few seconds.

Step7: Once the simulation is finished, we can see a new data set has been added under Workspace Tree "MonteCarlo1_Data". Double click on this data to open it. Right click on **BeamMeas_PhiCutDirectivity** to plot it on a new graph. We can clearly see the variation in beam pattern in Phi Cut due to 10% Element Failure.



Lab 10: Antenna Active Loading Analysis

Objective of this lab is to understand how to take care of active loading which is pretty important consideration in a Phase Array system.

Step1: Open Lab5_RF_Phased_Array and save it as **Lab10_Antenna_ActiveLoading**

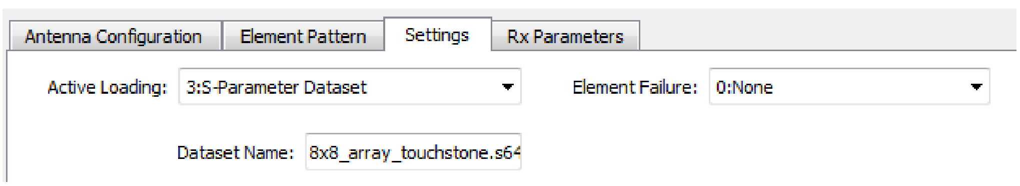
Step2: Copy/Paste Design1 and PhasedArray1. **Rename** the copied schematic as ActiveLoading and Simulation Controller as PhasedArray_ActiveLoading.

Step3: Double click on PhasedArray_ActiveLoading select design as ActiveLoading

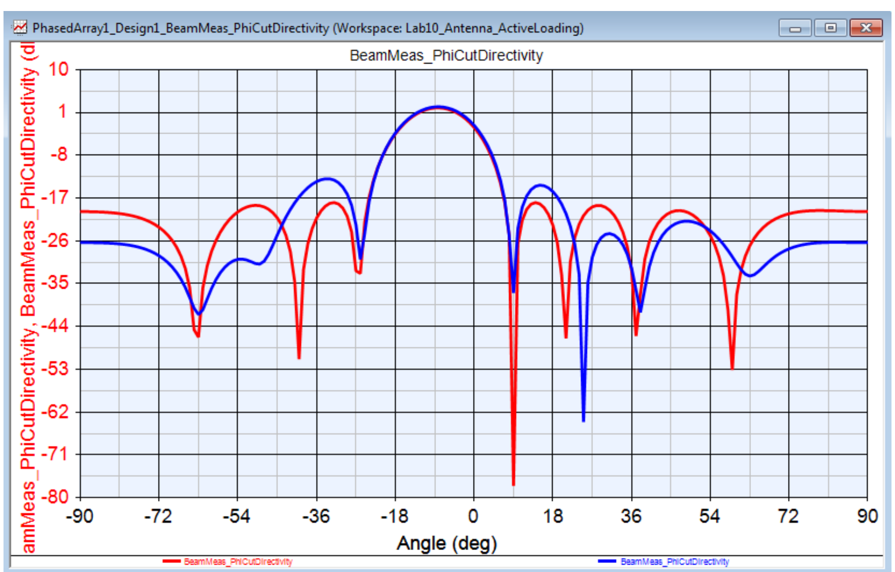
Step4: Right click on Designs folder, select Import->S-Data file… and select file: **8x8_array_touchstone.s64p** from the provided workshop folder.

Note: *An 8x8 patch antenna array was solved in the ADS Momentum field solver, and the S-parameters stored to a file (*8x8_array_touchstone.s64p*), which we imported into this workspace. Each of the 64 patches can "talk" to the other 63 patches, resulting in a 64x64 S-matrix at one frequency, at one scan angle (4096 values!). This is much more accurate, and provides a physical basis for the coupling. But these matrixes can get large.*

Step5: Open ActiveLoading schematic, double click on ArrayAntenna component. Under Setting tab, select Active Loading as S-Parameter Dataset and enter dataset name as: **8x8_array_touchstone.s64p** (file name which we imported)



Step6: Run Simulation, open the dataset, select BeamMeas_PhiCutDirectivity and plot it on earlier rectangular graph with 2D cut.



Notice the impact of Active loading of antenna elements in the resulting antenna pattern for Phased Array system

Lab 11: Importing EMPro/HFSS radiation pattern file

Objective: In this lab, we will learn how to import EMPro/HFSS/CST radiation pattern file into SystemVue Phased Array system simulation

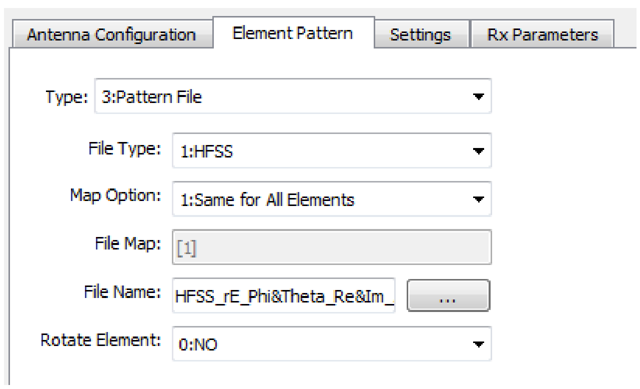
Step1: Open Lab5_RF_Phased_Array and save it as **Lab11_Antenna_HFSSImport**

Step2: Copy/Paste Design1 and PhasedArray1. **Rename** the copied schematic as HFSSPattern and Simulation Controller as PhasedArray_HFSS.

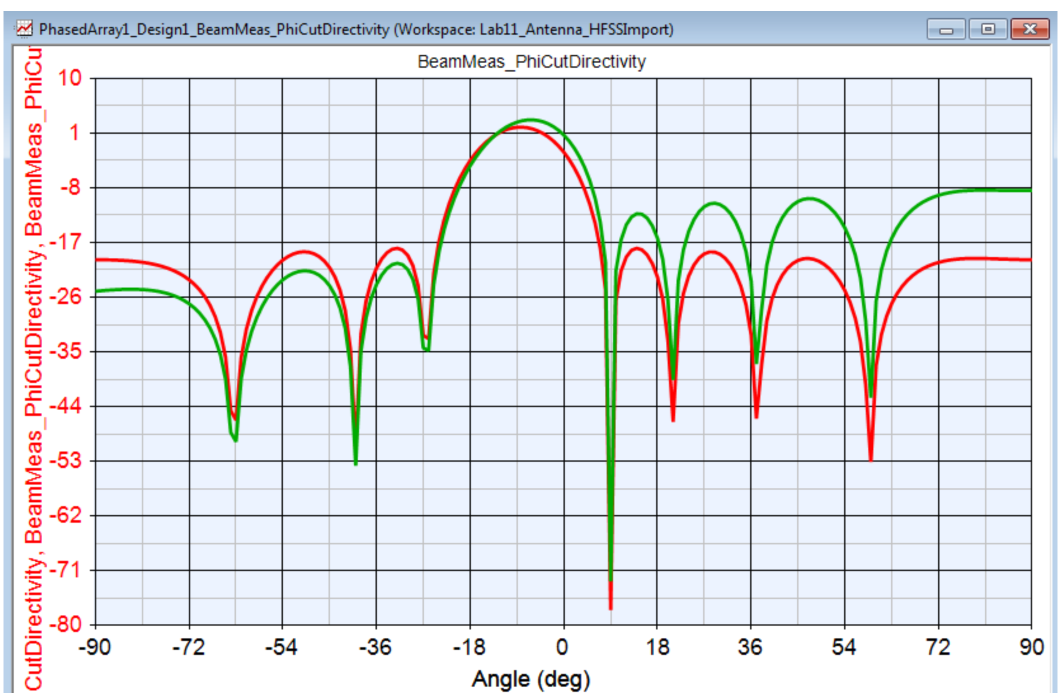
Step3: Double click on PhasedArray_HFSS select design as **HFSSPattern**

Step4: Open HFSSPattern, double click on ArrayAntenna component. Under Element Pattern:

- Type = Pattern File
- File Type = HFSS (note that EMPro, HFSS and CST formats are supported)
- Map Option = Same for All Elements (we can have different pattern file for different radiators)
- File Name = HFSS_rE_Phi&Theta_Re&Im_ANT001.csv (available under provided workshop folder)



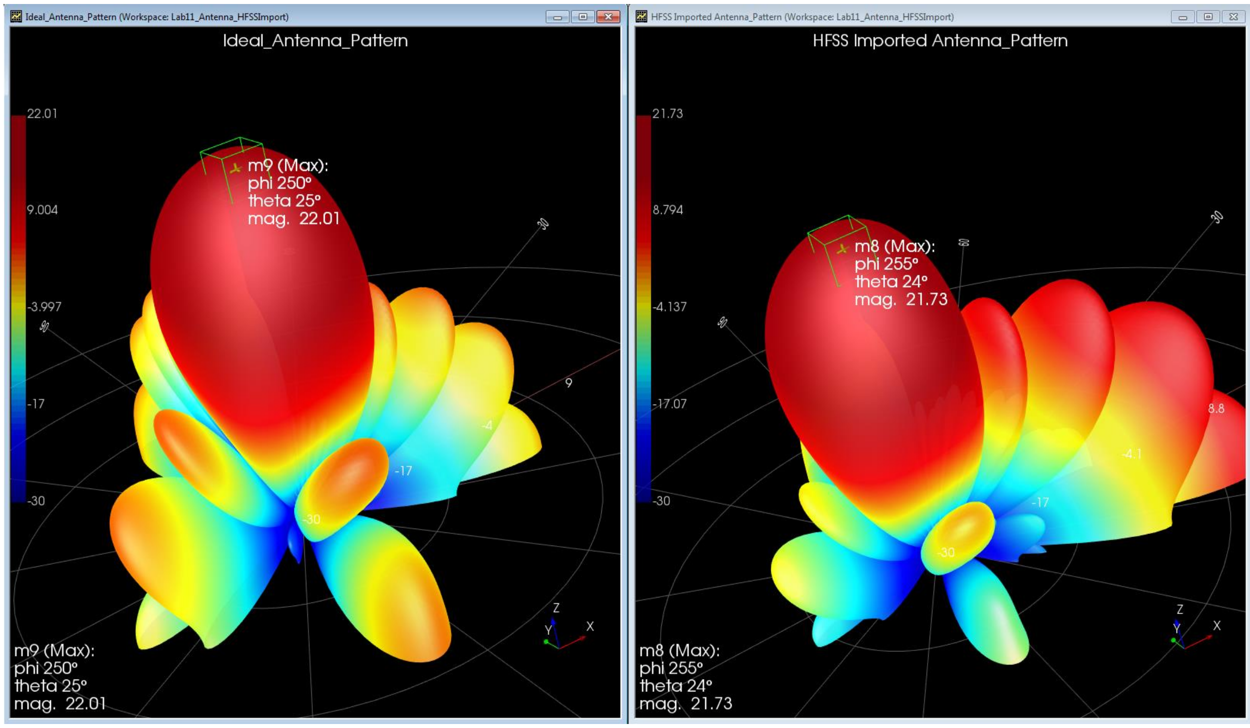
Step5: Run simulation, open dataset, plot BeamMeas_PhiCutDirectivity onto the earlier rectangular 2D-cut plot



Step6: Open 3D Plot equations page, add following lines to existing plot code:

```
using('PhasedArray_HFSS_HFSSPattern_Data');

graph1=graph_figure('HFSS Imported Antenna_Pattern');

graph1.add_spherical_data( ArrayMeas_Directivity_PhiGrid , ArrayMeas_Directivity_ThetaGrid,
max(10*log10(ArrayMeas_Directivity_Pattern),-30) );
```

Step7: Run equations and open both the 3D plots and notice the difference between Isotropic based radiator element and HFSS designed radiator element based antenna pattern. Right click place Marker on Maximum point to also notice that beam pointing has error of 1 deg in Theta and 5 deg in Phi.



Note on Element Pattern File:

- **ElementPatternType** parameter specifies the antenna pattern for each element in the phased array. This parameter can be set to one of the options below:

  - **0: Isotropic**
    Each array element is assumed to be a theoretical isotropic antenna that radiates uniformly in all directions.

  - **1: Three Sector Antenna for Y-Z URA**
    The antenna element pattern is defined based on [1] and it only supports uniform rectangular array (URA) positioned on the Y-Z plane. This option is only available in data flow domain.

  - **2: Cosine Antenna**
    The antenna element pattern is defined as ep(θ, ϕ) = $\cos^N(\theta)$, where ϕ and θ are the angle

coordinates of the reference spherical coordinate system and *N* is the parameter **CosineExponent**.

- **3: Pattern File**
  Antenna pattern files generated by EMPro, HFSS and CST can be imported for more realistic modeling.

  - **ElementPatternFileType**
    ElementPatternFileType specifies which element pattern file format to use. When ElementPatternFileType = *EMPro*, the pattern format needs to be in EMPro format. When ElementPatternFileType = *HFSS*, the pattern format needs to be in one of the supported HFSS formats described below. When ElementPatternFileType = *CST*, the pattern format needs to be in one of the supported CST formats described below.

  - **ElementPatternFileMapOption**
    ElementPatternFileMapOption specifies how to map element pattern files to antenna elements defined in Phased Array Configuration Parameters. It supports the following options:

    - **0: Individual for Each Element**
      Each antenna element corresponds to individual antenna pattern file and the file name must contain the index number at the end, for example: Antenna001.uan, Antenna002.uan, ..., Antenna256.uan. The **ElementPatternFileName** should specify the first element pattern file name, i.e., Antenna001.uan in the aforementioned example.

    - **1: Same for All Elements**
      All the antenna elements share the same antenna pattern file specified using **ElementPatternFileName** parameter.

    - **2: Custom**
      The Custom option allows you to map M number of element pattern files to N antenna elements, where M <= N. You can use **ElementPatternFileMap** parameter to map a subset of element pattern files to entire array, the element value in **ElementPatternFileMap** is the index of the pattern file.

      For Uniform Rectangular Array and Rectangular Array with Triangular Grid configurations, the dimensions of **ElementPatternFileMap** must divide the dimensions of the 2D matrix exactly For Custom, Uniform Linear Array, Circular, and Uniform Hexagonal Array configurations, **ElementPatternFileMap** must be vectors (1 x N or N x 1) and N must divide the total number of antenna elements exactly.

  When NumAntennasPerLocation = 2, ElementPatternFileMap must have an even number of rows (the second antenna element at the same location is treated as a new row). For example, given a URA with NumElementsX = 2 and NumElementsY=2 and NumAntennasPerLocation = 2, there will be total 8 antennas represented in a 4x2 matrix. In this case, ElementPatternFileMap = [1;2] will assign element pattern files 1 and 2 to the two antennas in each location, which is equivalent to ElementPatternFileMap = [1, 1; 2, 2; 1, 1; 2, 2].

- **4: Pattern Data**

  All elements in the array use the same element pattern defined using the following parameters:
  - **PatternDataThetaGrid**: Theta grid in M x N matrix. M is the number of phi angles and N is the number of theta angles. Theta should be [0,180], and the theta interval should be consistent.
  - **PatternDataPhiGrid**: Phi grid in M x N matrix. M is the number of phi angles and N is the number of theta angles. Phi should be [0,360], and the phi interval should be consistent.
  - **PatternDataEtheta**: Complex Etheta in M x N matrix. M is the number of phi angles and N is the number of theta angles.
  - **PatternDataEphi**: Complex Ephi in M x N matrix. M is the number of phi angles and N is the number of theta angles.

Note: ***To examine a particular element pattern, you can set array size to 1x1, set ElementPatternType = Pattern File, set ElementPatternFileMapOption to Same for All Elements, and specify the element pattern file in ElementPatternFileName.***

For HFSS, SystemVue supports the following formats:

Format 1	Phi[deg]	Theta[deg]	mag(GainPhi) or dB(GainPhi)	mag(GainTheta) or dB(GainTheta)	Any other fields will be ignored		
Format 2	Phi[deg]	Theta[deg]	Mag(RealizedGainPhi) or dB(RealizedGainPhi)	mag(RealizedGainTheta) or dB(RealizedGainTheta)			
Format 3	Phi[deg]	Theta[deg]	mag(rEPhi)[xx]	mag(rETheta)[xx]	ang_deg(rEPhi)	ang_deg(rETheta)	Any other fields will be ignored
Format 4	Phi[deg]	Theta[deg]	re(rEPhi)	im(rEPhi)	re(rETheta)	im(rETheta)	Any other fields will be ignored
Format 5	Phi[deg]	Theta[deg]	dB(GainTotal)	Should NOT contain other fields			

For format 1 and format 2, the gain defined in the file will be used directly as it already contains the radiation efficiency (we suggest using RealizedGainTheta and RealizedGainPhi as it's more realistic) For format 3, format 4 and format 5, SystemVue can only calculate the directivity of each antenna element, and radiation efficiency is assumed to be 1.

For CST, SystemVue supports the following format:

Phi	*Theta*	*Re(E_Theta)*	*Im(E_Theta)*	*Re(E_Phi)*	*Im(E_Phi)*	*Any other fields will be ignored*

For all formats, theta should be [0, 180], and phi should be [0,360]. Theta interval in the pattern file should be consistent and > 0. Phi interval in the pattern file should be consistent and > 0.