Workshop: Top-Down Design of an RF Receiver and Antenna Front-End Integration

This example designs a RF receiver for a ZigBee®-like application using a top-down methodology. It verifies the error rate performance after the addition of impairments such as interfering signals and the integration of the antenna front-end. The example uses the `RF Budget Analyzer` App to rank the elements contributing to the noise and non-linearity budget, and the `Antenna Designer App` to design the antenna.

These are the specifications of the ZigBee-like system and the high-level specifications of the receiver:

- Data rate = 250kbps
- OQPSK modulation with half sine pulse shaping, as specified in IEEE® 802.15.4 for the physical layer of ZigBee
- Direct sequence spread spectrum with chip rate = 2Mchips/s
- Sensitivity specification = -100dBm
- Bit Error Rate (BER) specification = 1e-4
- Analog to digital converter (ADC) with 10 bits and 0dBm saturation power

Based on the above specifications, this example shows how to use design heuristics to identify the desired Noise Figure and Gain for the RF receiver:

- **NF = 10.7dB**
- **G = 51.5dB**

This workshop guides you through the following exercises:

Exercise 1:

- Starting from the receiver Gain and Noise Figure, design and analyze a cascade of RF components and integrate them in the system-level model
- Measure the corresponding Chip Error Rate (ChER) in presence of an out-of-band interfering signal

Exercise 2:

- Integrate a simple dipole antenna into the system-level model of the RF receiver
- Understand the impact of polarization mismatch

Exercise 3:

- Design a dual polarized antenna, analyze its performance using EM analysis, and integrate it in the system-level model of the RF receiver
- Design a Wilkinson combiner and integrate it together with the dual polarized antenna into the system-level model

Exercise 4:

- Design a 90-degrees phase shifter and implement it on a PCB together with the Wilkinson combiner

- Analyze the PCB structure using electromagnetic analysis
- Verify the system performance in terms of ChER

Before you start with the exercises, let's add the folder with the solution ot the path to allow verify each step.

```
addpath('./Solutions')
```

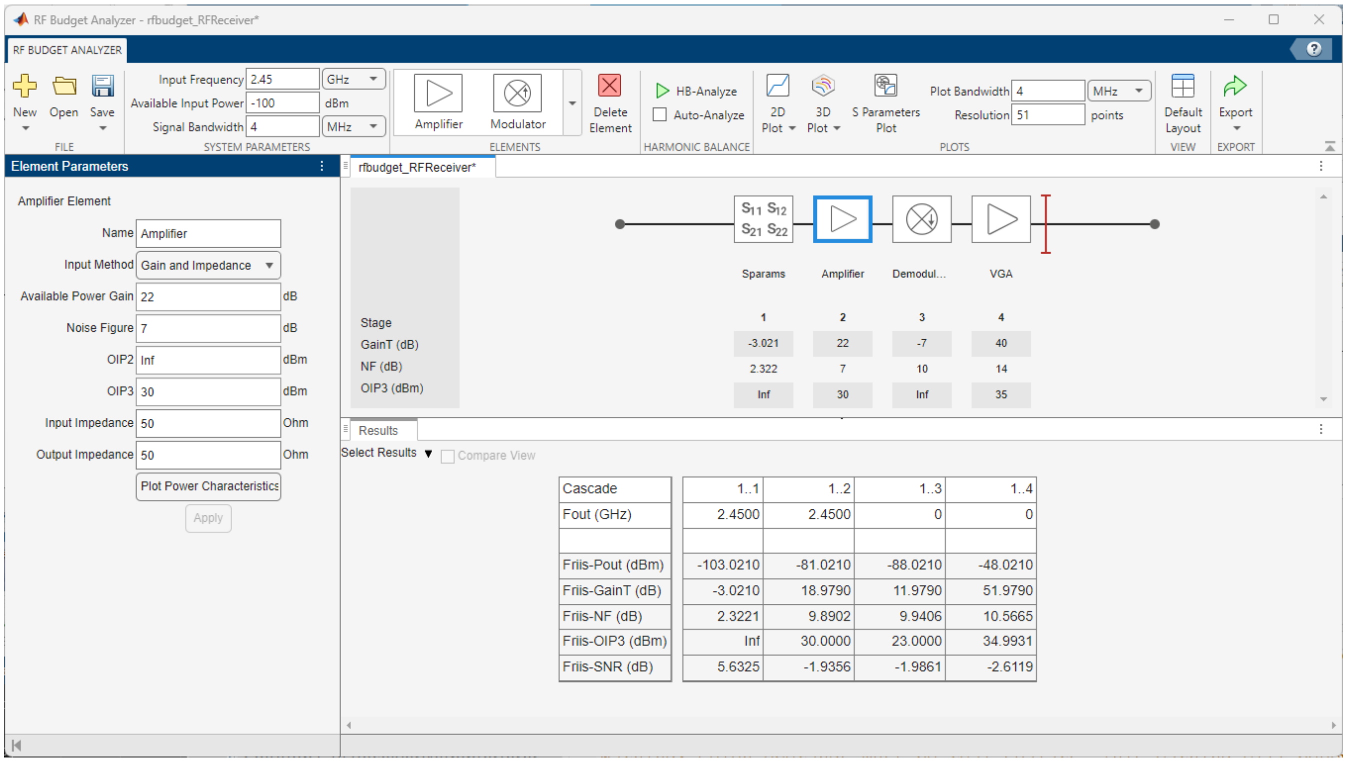
Exercise 1: Design of RF Receiver

Number of steps: 6

Expected completion time: 30 minutes

In this section, the RF receiver with the desired noise figure and gain specifications is implemented using four discrete sub-components with these characteristics:

- S-parameters SAW filter described by means of the Touchstone file *SAWfilter.s2p*
- Amplifier LNA with Gain = 22 dB, Noise Figure = 7 dB, OIP3 = 30dBm
- Demodulator with Local Oscillator = 2.45GHz, Gain = -7 dB, OIP2 = 55dBm, Noise Figure = 10 dB
- Amplifier VGA with Gain = 40 dB, Noise Figure = 14 dB, OIP3 = 35dBm



Input Frequency: 2.45 GHz
Available Input Power: -100 dBm
Signal Bandwidth: 4 MHz
Plot Bandwidth: 4 MHz
Resolution: 51 points

Amplifier Element

Name	Amplifier	
Input Method	Gain and Impedance	
Available Power Gain	22	dB
Noise Figure	7	dB
OIP2	Inf	dBm
OIP3	30	dBm
Input Impedance	50	Ohm
Output Impedance	50	Ohm

Stage	1	2	3	4
GainT (dB)	-3.021	22	-7	40
NF (dB)	2.322	7	10	14
OIP3 (dBm)	Inf	30	Inf	35

Cascade	1..1	1..2	1..3	1..4
Fout (GHz)	2.4500	2.4500	0	0
Friis-Pout (dBm)	-103.0210	-81.0210	-88.0210	-48.0210
Friis-GainT (dB)	-3.0210	18.9790	11.9790	51.9790
Friis-NF (dB)	2.3221	9.8902	9.9406	10.5665
Friis-OIP3 (dBm)	Inf	30.0000	23.0000	34.9931
Friis-SNR (dB)	5.6325	-1.9356	-1.9861	-2.6119

1.1 Launch at the command line the RF Budget Analyzer app and construct a chain as shown above.

```
rfBudgetAnalyzer
```

1.2 Compare budget results obtained with Friis equations and Harmonic Balance analysis and verify the noise, gain, SNR budget of the RF chain for the ZigBee-like signal with:

- center frequency = 2.45GHz
- available input power = -100dBm
- bandwidth = 4MHz

Verify the budget results obtained with Friis equations and Harmonic Balance analysis for the interfering signal:

- center frequency = 2.5GHz
- available input power = -30dBm
- bandwidth = 4MHz

What can you say about the impact of the interfering signal on the RF receiver?

1.3 From the RF Budget Analyzer app generate an RF Blockset Circuit Envelope receiver model. Integrate the generated model in the available placeholder in the testbench `Exercise1_RFReceiverWithInterferer.slx`. Spend a few minutes to familiarize yourself with testbench. In particular, inspect the baseband signal generation, the ADC, and the baseband receiver.

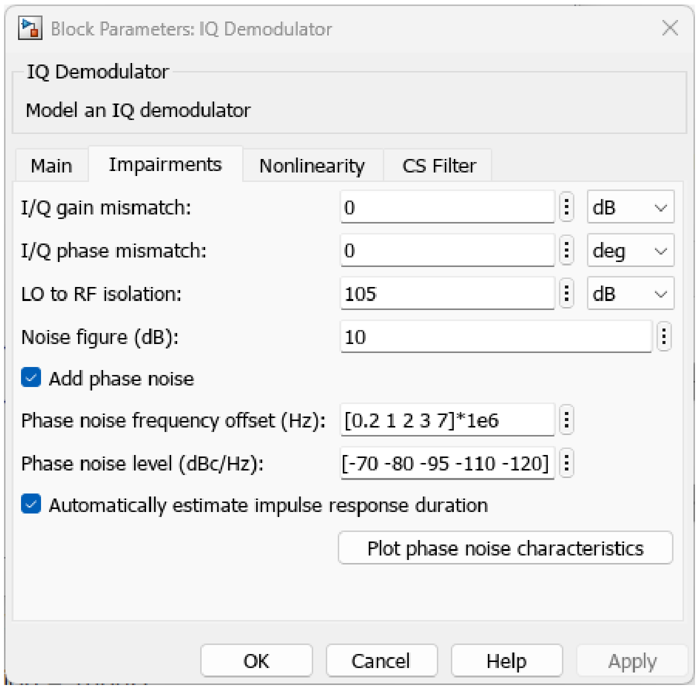
```
open_system('Exercise1_RFReceiverWithInterferer.slx')
```

1
RF In
Zs
SL
RF
RF
BB
Re(u)
I Out
1
BB
Re(u)
Q Out
2

Why is the demodulator implemented with an quadrature (IQ) architecture?

1.4 Open the mask of the IQ demodulator block, and add the following impairments:

- LO to RF isolation = 105dB
- Phase noise frequency offset = [0.2,1,2,3,7] MHz
- Phase noise levels = [-70,-80,-95,-110,-120] dBc/Hz


Block Parameters: IQ Demodulator

IQ Demodulator

Model an IQ demodulator

I/Q gain mismatch:	0 dB
I/Q phase mismatch:	0 deg
LO to RF isolation:	105 dB
Noise figure (dB):	10
✓ Add phase noise	
Phase noise frequency offset (Hz):	[0.2 1 2 3 7]*1e6
Phase noise level (dBc/Hz):	[-70 -80 -95 -110 -120]
✓ Automatically estimate impulse response duration	

1.5 Open the input port, and inspect its parameters.

How many incoming signals are entering the receiver?

1.6 Inspect the spectrum of the outptut signal and verify the ChER in the following scenarios:

- Interfering signal power = -30dBm with DC offset correction
- Interfering signal power = -200dBm, with and without DC offset correction

What the cause of the residual DC offset?

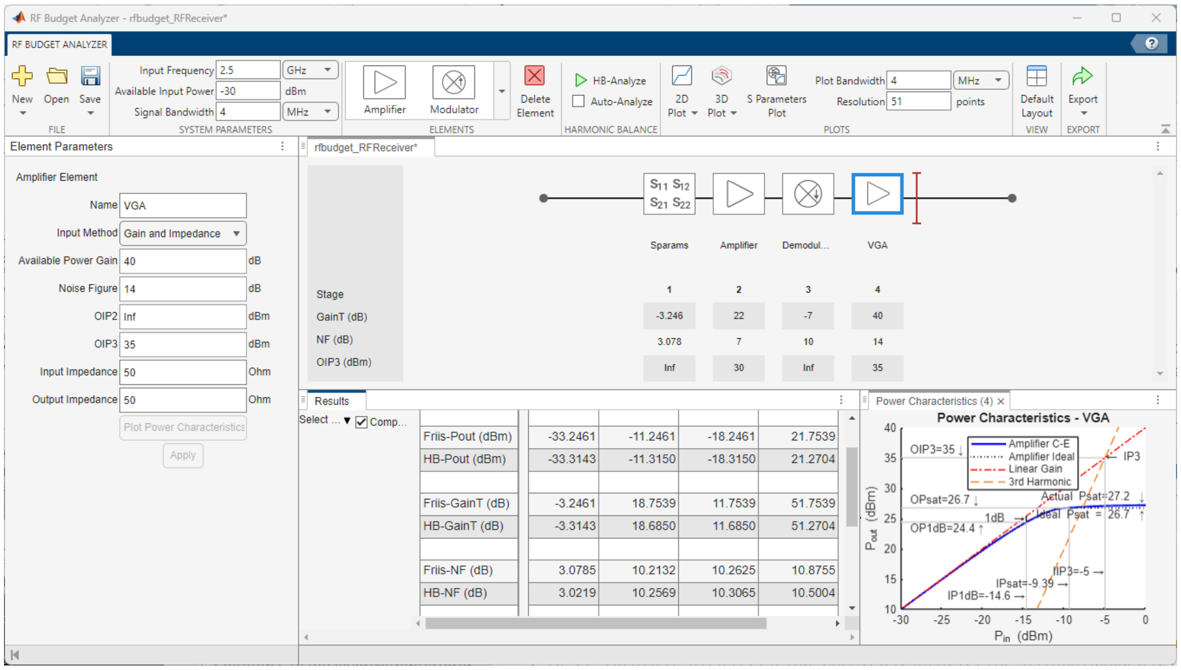
Solution

Execute the command below to visualize the RF receiver chain in the **RF Budget Analyzer** app.

```
rfBudgetAnalyzer('rfbudget_RFReceiver.mat')
```

The interfering signal has higher power than the desired signal, and it is entering the saturation region of the VGA. Although the interfering signal is centered around a different frequency, saturation effects will impact also the desired signal and will ultimately result into higher ChER.

This can be visualized by plotting the power characteristic of the VGA amplifier either in the budget app or in the block mask.



Input Frequency	2.5	GHz
Available Input Power	-30	dBm
Signal Bandwidth	4	MHz
Plot Bandwidth	4	MHz
Resolution	51	points

Amplifier Element

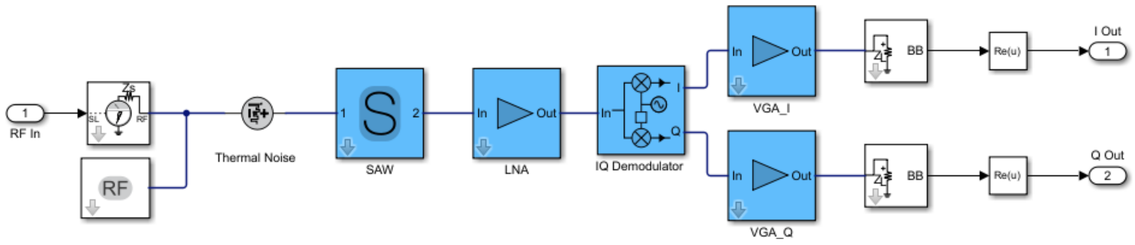
Name	VGA	
Input Method	Gain and Impedance	
Available Power Gain	40	dB
Noise Figure	14	dB
OIP2	Inf	dBm
OIP3	35	dBm
Input Impedance	50	Ohm
Output Impedance	50	Ohm

Stage	1	2	3	4
GainT (dB)	-3.246	22	-7	40
NF (dB)	3.078	7	10	14
OIP3 (dBm)	Inf	30	Inf	35

Friis-Pout (dBm)	-33.2461	-11.2461	-18.2461	21.7539
HB-Pout (dBm)	-33.3143	-11.3150	-18.3150	21.2704
Friis-GainT (dB)	-3.2461	18.7539	11.7539	51.7539
HB-GainT (dB)	-3.3143	18.6850	11.6850	51.2704
Friis-NF (dB)	3.0785	10.2132	10.2625	10.8755
HB-NF (dB)	3.0219	10.2569	10.3065	10.5004

Open the model where the RF receiver has been integrated:

```
open_system('Solution1_RFReceiverWithInterferer.slx');
```



As the receiver implements a direct conversion architecture, a quadrature demodulator is required to recover a complex modulated signal.

Two signals are entering the receiver: the desired signal centered at 2.45GHz, and a high power interfering signal at 2.5GHz.

The interfering signal is exciting the non-linearity of the demodulator specified by means of IP2. In absence of a high-power signal, the residual DC offset is caused by finite LO to RF isolation.

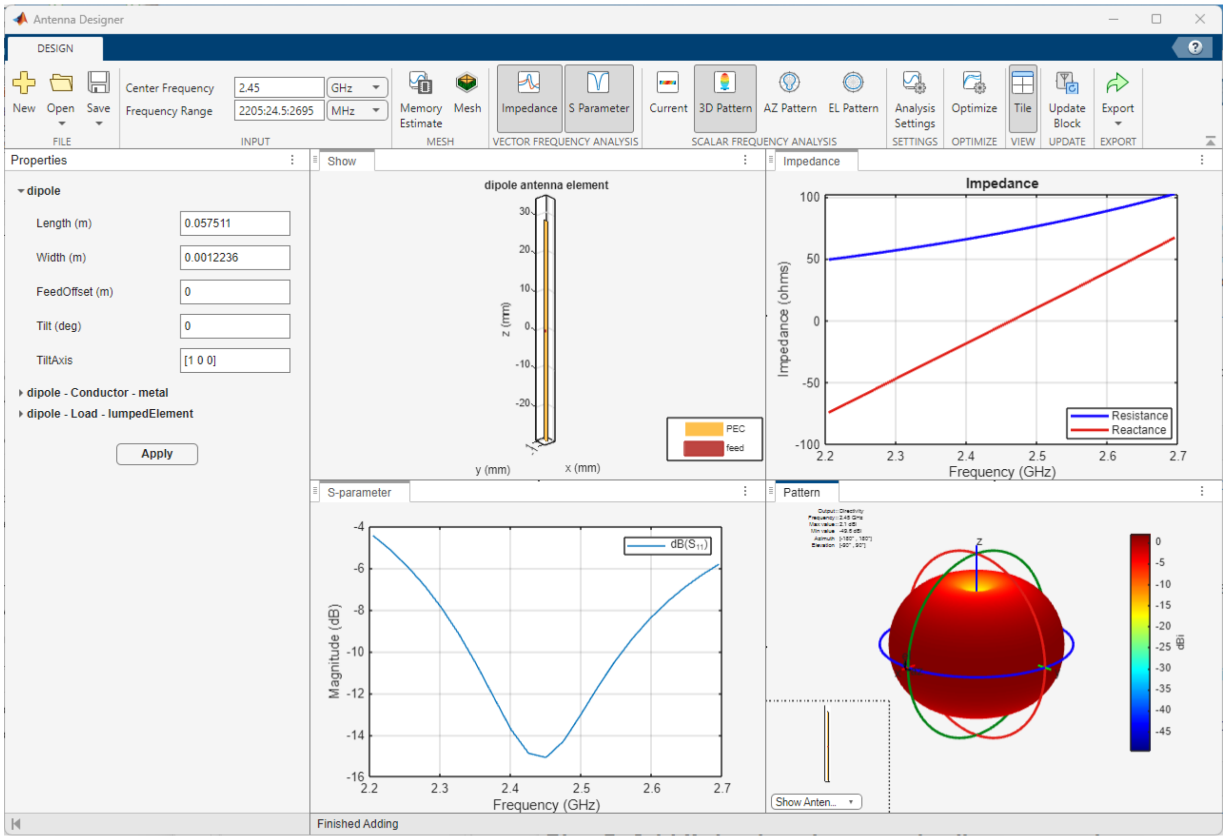
Extra: use the RF budget analysis of the chain to inspect the S-parameters of the cascade, and verify the phase angle of the last stage. Open the baseband receiver model, and see how this angle has been used to align the received constellation.

Exercise 2: Integration of RF Receiver Antenna

Number of steps: 3

Expected completion time: 15 minutes

In this exercise, you integrate a simple dipole antenna in the RF receiver and experiment with changing the polarization of the incoming signals.


Center Frequency 2.45 GHz
Frequency Range 2205:24.5:2695 MHz

dipole

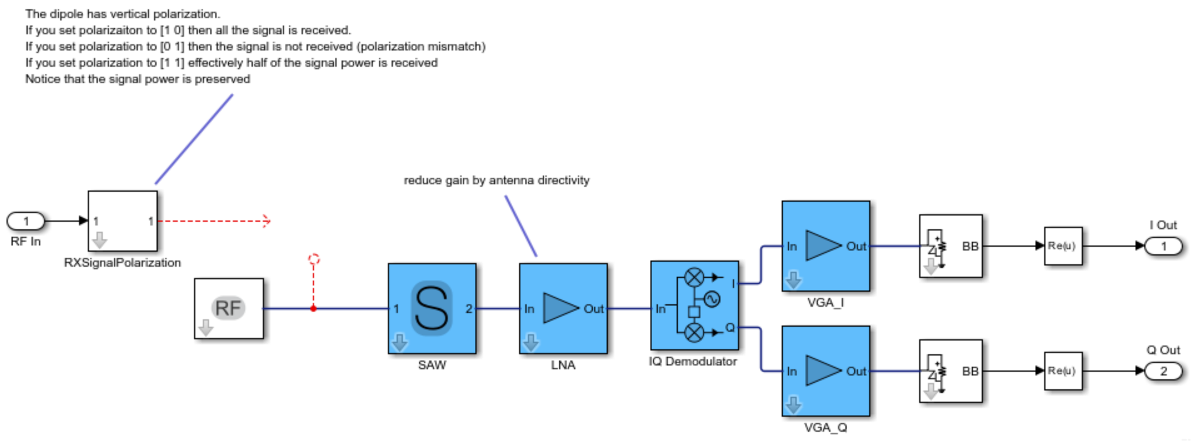
Length (m) 0.057511
Width (m) 0.0012236
FeedOffset (m) 0
Tilt (deg) 0
TiltAxis [1 0 0]

dipole - Conductor - metal
dipole - Load - lumpedElement

Finished Adding

2.1 Open the Simulink testbench `Exercise2_RFReceiverWithAntenna`, and open the RF receiver subsystem. Notice that the input port is missing.

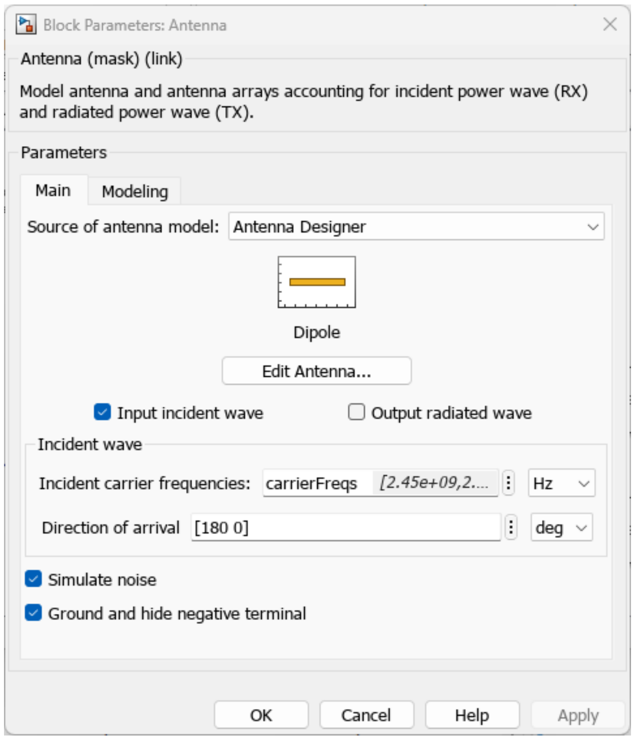
```
open_system('Exercise2_RFReceiverWithAntenna.slx')
```


The dipole has vertical polarization.
If you set polarizaiton to [1 0] then all the signal is received.
If you set polarization to [0 1] then the signal is not received (polarization mismatch)
If you set polarization to [1 1] effectively half of the signal power is received
Notice that the signal power is preserved

reduce gain by antenna directivity

2.2 Insert an antenna block from the RF Blockset Circuit Envelope library. You can either open the Simulink library browser, or simply type "antenna" in the Simulink canvas. Open the mask of the antenna block and configure it as following:

- the antenna should operate in receiving mode
- the incident carrier frequencies should be set by the variable `carrierFreqs` defined in the workspace (make sure that you change the units to Hz)
- set the source of the antenna model to be `Antenna Designer` and click on the button `Create Antenna` to open the Antenna Designer app
- in the Antenna Designer App, design a dipole and analyze its S-parameters and far-field radiation pattern as shown in the image above. Click on `Update Block` and return to the Simulink testbench


Block Parameters: Antenna

Antenna (mask) (link)

Model antenna and antenna arrays accounting for incident power wave (RX) and radiated power wave (TX).

Parameters

Main | Modeling

Source of antenna model: Antenna Designer

Dipole

Edit Antenna...

☑ Input incident wave ☐ Output radiated wave

Incident wave

Incident carrier frequencies: carrierFreqs [2.45e+09,2.... Hz

Direction of arrival [180 0] deg

☑ Simulate noise

☑ Ground and hide negative terminal

OK Cancel Help Apply

2.3 Connect the antenna block to the RF Receiver. Open the mask parameters of the subsystem `RXSignalPolarization`, and verify the ChER in the following scenario:

- Polarization Interferer = [1 0], with and without DC offset removal
- Polarization Interferer = [0 1], with and without DC offset removal

Why is the ChER still acceptable even without DC offset correction, when the polarization of the interfering signal lies entirely on the horizontal plane?

Solution

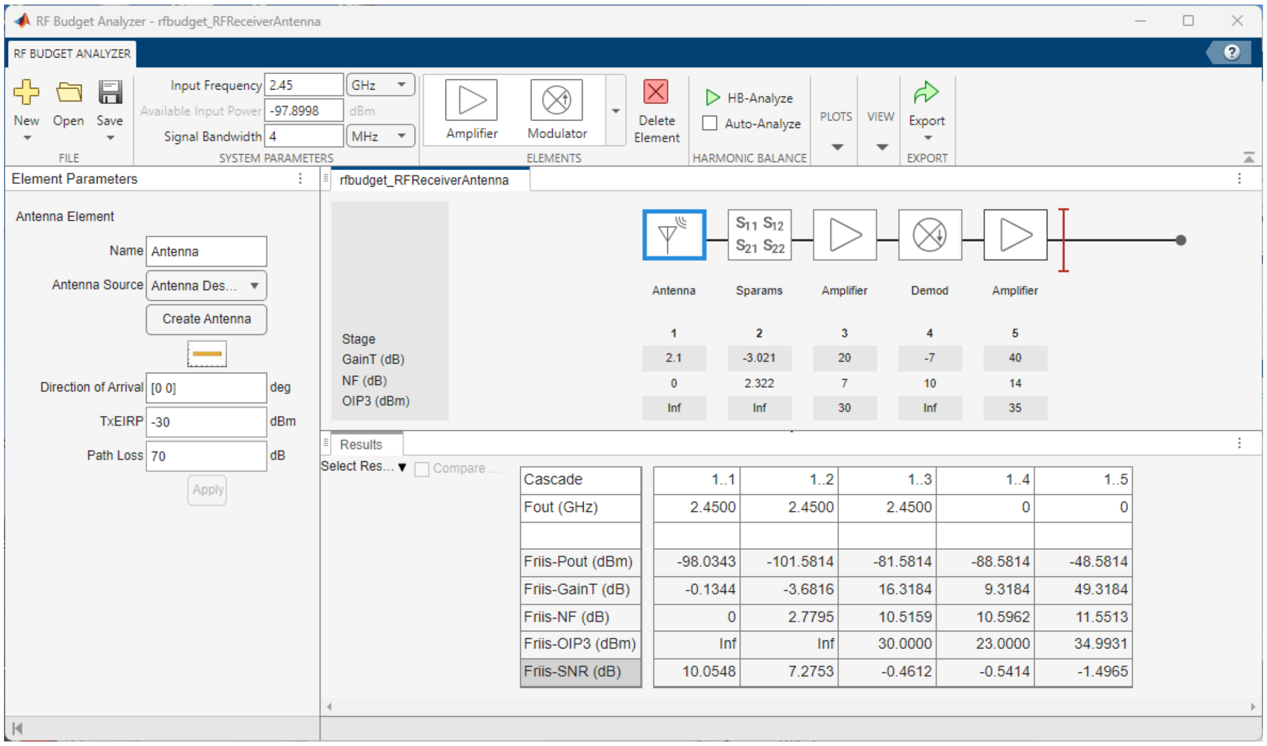
Open the resulting integrated model:

```
open_system('Solution2_RFReceiverWithAntenna.slx')
```

The dipole is a linearly polarized antenna along the vertical plane, hence incoming signals with horizontal polarization are not received.

Extra: you can verify the updated budget including the antenna block. Notice that to maintain the output power consistent with the initial budget, the amplifier gain was reduced by 2dB, the same amount as the antenna directivity gain.

```
rfBudgetAnalyzer('rfbudget_RFReceiverAntenna.mat')
```



Input Frequency 2.45 GHz
Available Input Power -97.8998 dBm
Signal Bandwidth 4 MHz

Antenna Element

Name: Antenna
Antenna Source: Antenna Des...
Create Antenna
Direction of Arrival: [0 0] deg
TxEIRP: -30 dBm
Path Loss: 70 dB
Apply

rfbudget_RFReceiverAntenna

	Antenna	Sparams	Amplifier	Demod	Amplifier
Stage	1	2	3	4	5
GainT (dB)	2.1	-3.021	20	-7	40
NF (dB)	0	2.322	7	10	14
OIP3 (dBm)	Inf	Inf	30	Inf	35

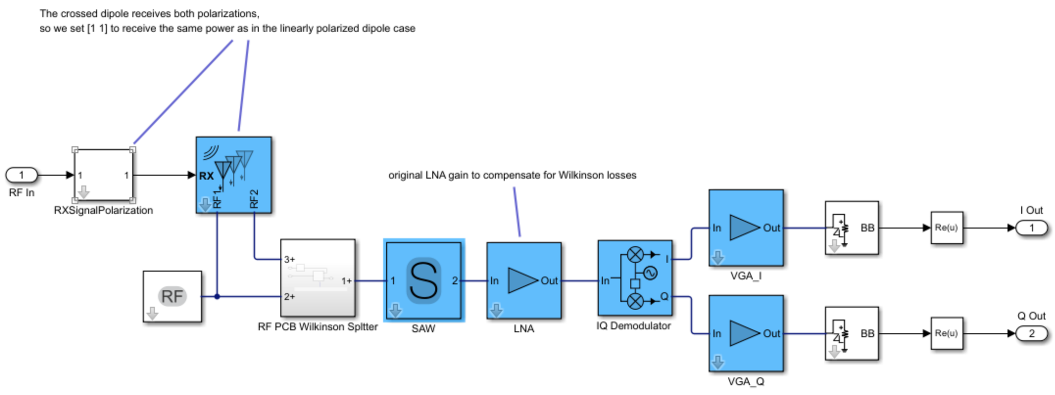
Cascade	1..1	1..2	1..3	1..4	1..5
Fout (GHz)	2.4500	2.4500	2.4500	0	0
Friis-Pout (dBm)	-98.0343	-101.5814	-81.5814	-88.5814	-48.5814
Friis-GainT (dB)	-0.1344	-3.6816	16.3184	9.3184	49.3184
Friis-NF (dB)	0	2.7795	10.5159	10.5962	11.5513
Friis-OIP3 (dBm)	Inf	Inf	30.0000	23.0000	34.9931
Friis-SNR (dB)	10.0548	7.2753	-0.4612	-0.5414	-1.4965

Exercise 3: Integrate a crossed polarized dipole antenna and a Wilkinson combiner

Number of steps: 3

Expected completion time: 15 minutes

In this section you design a dipole crossed antenna and a Wilkinson combiner to merge the two polarization components of the incoming signal. Notice that the feed network of the antenna must implement a 90 degrees relative phase shift. The antenna and the Wilkinson combiner are integrated in the RF receiver.


The crossed dipole receives both polarizations,
so we set [1 1] to receive the same power as in the linearly polarized dipole case
original LNA gain to compensate for Wilkinson losses
RF In
RXSignalPolarization
RX
RF1
RF2
RF
RF PCB Wilkinson Splitter
SAW
LNA
IQ Demodulator
VGA_I
VGA_Q
BB
Re(u)
I Out
Q Out
In
Out

3.1 Design the Antenna Toolbox antenna catalog element `dipoleCrossed` to operate at the desired center frequency. Visualize the far field radiation pattern and the S-parameters.

```
Xdipole = design(dipoleCrossed, 2.45e9);
Xdipole.Tilt = 45;
figure; show(Xdipole);
figure; pattern(Xdipole,2.45e9);
sparamXdipole = sparameters(Xdipole, (2205:24.5:2695)*1e6);
figure; rfplot(sparamXdipole);
```

3.2 Design the RF PCB Toolbox catalog element `wilkinsonSplitter` to operate at the desired center frequency. Analyze the S-parameters and save them as a Touchstone file. The Wilkinson splitter in this receiver will be used as a signal combiner.

```
wilkCombiner=design(wilkinsonSplitter, 2.45e9);
show(wilkCombiner);
sparamWilkinson=sparameters(wilkCombiner, (2205:24.5:2695)*1e6);
figure; rfplot(sparamWilkinson);
rfwrite(sparamWilkinson,'wilkinsonSplitterSparams.s3p')
```

3.3 Open the Simulink testbench `Exercise3_RFReceiverWithXdipole` and verify the ChER. In order to properly combine the signals at the 2 antenna ports you need to include a 90 degrees phase shift on one of the combiner branches.

Implement the correct phase shift.

```
open_system('Exercise3_RFReceiverWithXdipole.slx')
```

What is the cause for the ChER degradation?

Solution

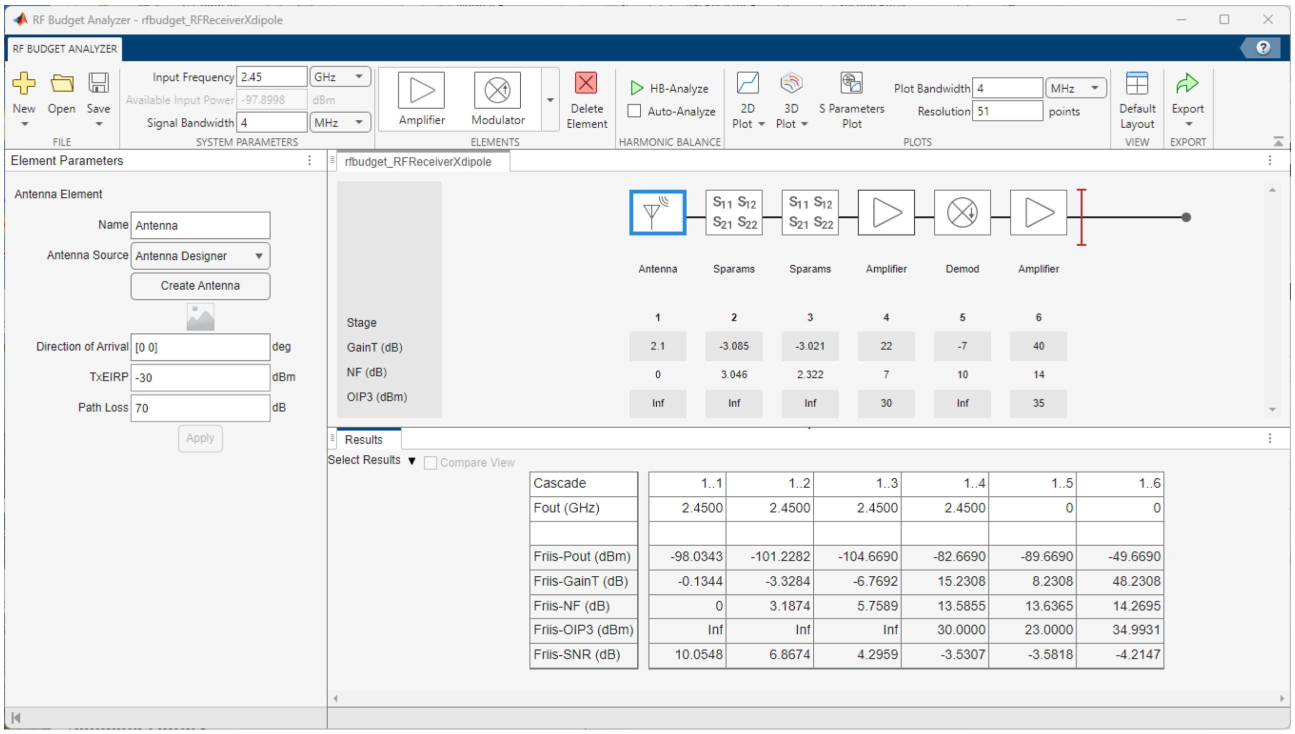
Open the model where the dipole crossed antenna and the Wilkinson combiner have been integrated together with a behavioral model of the phase shifter:

```
open_system('Solution3_RFReceiverWithXdipole.slx')
```

The additional insertion loss of the Wilkinson combiner contributes to additional noise and degrades the available SNR.

Extra: you can verify the updated budget including the losses of the Wilkinson combiner. For this, you need to identify the respective 2 port S-parameters of the combiner. Notice that the SNR is now reduced compared to the original budget and the total noise figure exceeds 14 dB.

```
rfwrite(snp2smp(sparamWilkinson,[1 2]),'wilkinson2ports.s2p')
rfBudgetAnalyzer('rfbudget_RFReceiverXdipole.mat')
```



Input Frequency 2.45 GHz
Available Input Power -97.8998 dBm
Signal Bandwidth 4 MHz
Plot Bandwidth 4 MHz
Resolution 51 points

Element Parameters

Antenna Element

Name: Antenna
Antenna Source: Antenna Designer
Direction of Arrival: [0 0] deg
TxEIRP: -30 dBm
Path Loss: 70 dB

rfbudget_RFReceiverXdipole

Stage	1	2	3	4	5	6
GainT (dB)	2.1	-3.085	-3.021	22	-7	40
NF (dB)	0	3.046	2.322	7	10	14
OIP3 (dBm)	Inf	Inf	Inf	30	Inf	35

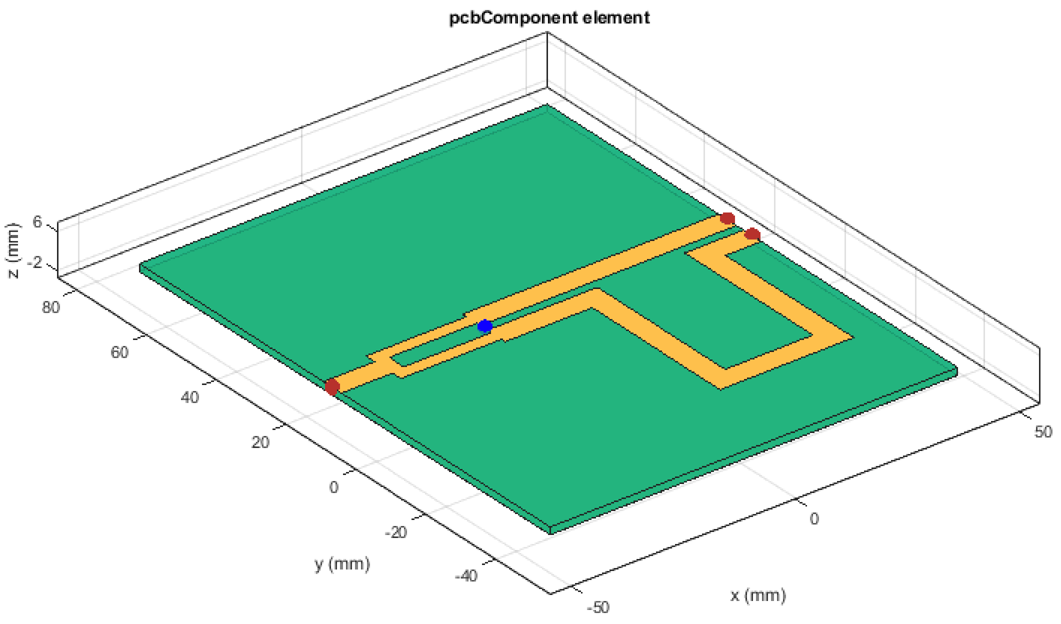
Cascade	1..1	1..2	1..3	1..4	1..5	1..6
Fout (GHz)	2.4500	2.4500	2.4500	2.4500	0	0
Friis-Pout (dBm)	-98.0343	-101.2282	-104.6690	-82.6690	-89.6690	-49.6690
Friis-GainT (dB)	-0.1344	-3.3284	-6.7692	15.2308	8.2308	48.2308
Friis-NF (dB)	0	3.1874	5.7589	13.5855	13.6365	14.2695
Friis-OIP3 (dBm)	Inf	Inf	Inf	30.0000	23.0000	34.9931
Friis-SNR (dB)	10.0548	6.8674	4.2959	-3.5307	-3.5818	-4.2147

Exercise 4: Design a Phase shifter and combine it on the PCB with the Wilkinson combiner

Number of steps: 6

Expected completion time: 15 minutes

In the previous step you determined that a 90 degrees was required to combine the two outputs from the antenna. In this step, you build the phase shifter using a distributed PCB element, and combine it on the PCB together with the Wilkinson splitter. You use EM analysis to determine the S-parameters and integrate them in the system-level model.



4.1 Design the phase shifter using the PCB catalog element `phaseShifter` to operate at the desired center frequency. Compute its S-parameters using EM analysis and verify that the implemented phase shift is 90 degrees.

The phase shifter will be appended on the PCB to one of the branches of the Wilkinson combiner.

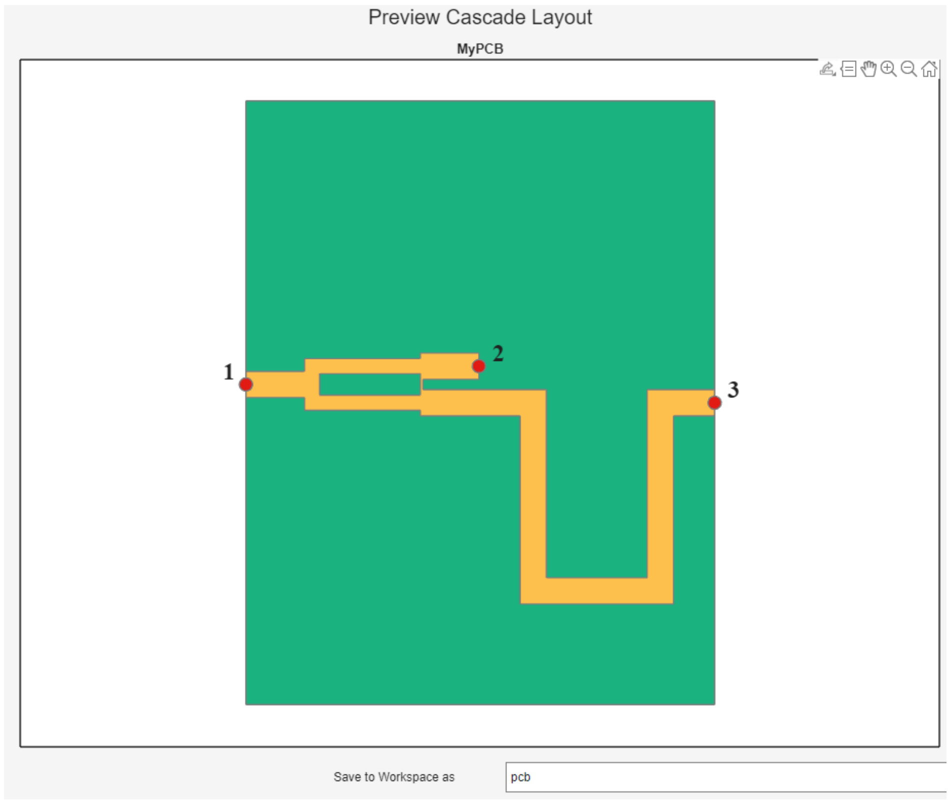
```
phaseShift = design(phaseShifter, 2.45e9);
figure; show(phaseShift);
sparamPhaseShift=sparameters(phaseShift, (2205:24.5:2695)*1e6);
figure; rfplot(sparamPhaseShift,'angle');
```

4.2 To maintain a rectangular shape for the PCB, you need to design a microstrip line with the same length as the phase shifter. This allows the other port of the Wilkinson splitter to reach the border of the board.

```
l1 = phaseShift.PortLineLength;
l2 = phaseShift.SectionShape.Length(2);
w = phaseShift.PortLineWidth;
mLine = microstripLine('Width',w, 'Length',2*(l1+w)+l2);
figure; show(mLine);
sparamLine=sparameters(mLine, (2205:24.5:2695)*1e6);
figure; rfplot(sparamLine,'angle');
```

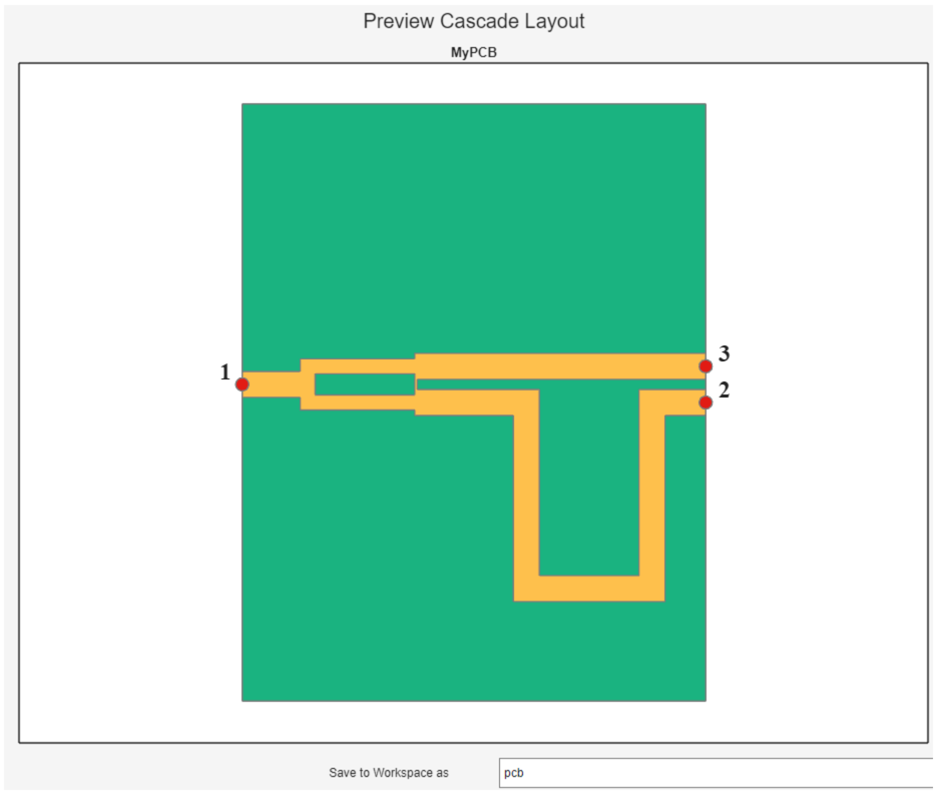
4.3 Use the graphical interface to connect port 3 of the Wilkinson splitter with port 1 of the phase shifter. Save the resulting object to the workspace with the name **pcb**.

```
pcbcascade(wilkCombiner,phaseShift,'Interactive',true);
```



4.4 Use again the same graphical interface to connect port 2 of the PCB structure created in the previous step with port 1 of the microstrip line. Save the resulting object to the workspace with the name **pcb**.

```
pcbcascade(pcb,mLine,'Interactive',true);
```



4.5 You use EM analysis to compute the S-parameters of the PCB, save them in a Touchstone file. Verify the 90 degrees phase difference along the 2 branches.

```
sparamWilkSplitterPhaseShift=sparameters(pcb, (2205:24.5:2695)*1e6);
figure; rfplot(sparamWilkSplitterPhaseShift,1,2,'angle');
hold on; rfplot(sparamWilkSplitterPhaseShift,1,3,'angle');
rfwrite(sparamWilkSplitterPhaseShift,'wilkinsonSplitterPhaseShiftSparams.s3p')
```

4.6 Update the Simulink model with the S-parameters of the PCB and run the simulation.

```
open_system('Exercise4_RFReceiverWithPCB.slx')
```

Extra: verify how the additional phase rotation introduced by the combiner / phase offset is accounted for in the baseband receiver and comment about:

- Is the model representative of the actual baseband receiver implementation?
- Would this approach also work for a wideband modulated signal?

Solution

Create the PCB integrating the Wilkinson combiner with the phase shift, the microstrip line. Compute the S-parameters and visualize them. Verify phase shifts on both branches as well as the transmission loss.

```
wilkSplitterPhaseShift = pcbcascade(wilkCombiner,phaseShift,3,1);
wilkSplitterPhaseShift = pcbcascade(wilkSplitterPhaseShift,mLine,2,1);
```

```
figure; show(wilkSplitterPhaseShift);
sparamWilkSplitterPhaseShift=sparameters(wilkSplitterPhaseShift,
(2205:24.5:2695)*1e6);
figure; rfplot(sparamWilkSplitterPhaseShift,1,2,'angle');
hold on; rfplot(sparamWilkSplitterPhaseShift,1,3,'angle');
figure; rfplot(sparamWilkSplitterPhaseShift,1,2);
rfwrite(sparamWilkSplitterPhaseShift,'wilkinsonSplitterPhaseShiftSparams.s3p')
```

Open the resulting integrated model:

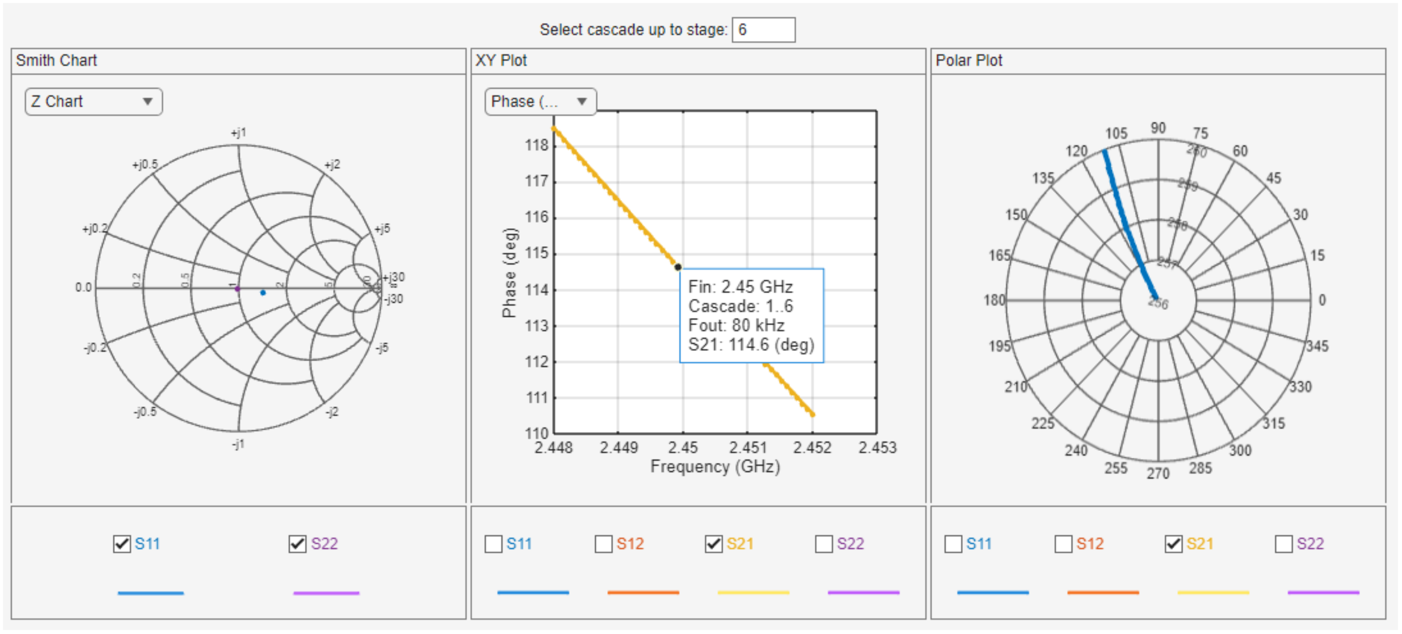
```
open_system('Solution4_RFReceiverWithPCB.slx')
```

The model is not representative of the actual baseband receiver implementation, as the relative phase shift is normally recovered by means of an adaptive clock and phase recovery system.

In this simple model, this approach could be used as the signal phase offset is constant and only dependent on the dispersive components in the chain. In case of a wideband modulated signal though the phase shift would be frequency dependent, and the receiver would require an adaptive equalizer.

The implemented phase shift of 115 degrees has been determined by inspecting the phase rotation in the budget analysis.

```
rfBudgetAnalyzer('rfbudget_RFReceiverXdipole.mat')
```


Select cascade up to stage: 6

Fin: 2.45 GHz
Cascade: 1..6
Fout: 80 kHz
S21: 114.6 (deg)

Conclusion

Following a top-down design methodology, you designed and analyzed the budget of an RF receiver using component specifications. After integrating an antenna, the impact of an interfering signal including polarization mismatch was simulated. A dual polarized antenna and the PCB feed network was integrated to increase modeling fidelity. The design was validated at each stage to confirm that the overall system performance goals were achieved.