

# How to Use S-parameter Models in PSpice

Author: Russell Carroll, PE Email: [russell@emisleuth.com](mailto:russell@emisleuth.com)

---

Accurate models of high frequency components are needed for signal integrity (SI) and electromagnetic compatibility (EMC) simulations. Developing parametric models, such as lumped element models, of these high frequency effects is difficult and have limited accuracy and usefulness. However, it is relatively easy to accurately quantify the high frequency effects with S-parameter models using a vector network analyzer (VNA). One disadvantage of S-parameter models is that they only describe the frequency response of the components and are difficult to use in commonly available analysis tools.

I have developed an advanced [S-parameter to PSpice Converter](#) tool along with an S-parameter for PSpice symbol library. These tools will allow anyone to accurately simulate multi-port S-parameter models in frequency domain and time domain simulations in a PSpice environment including the free [PSPICE-FOR-TI](#) program. This is critical for engineers that need accurate modeling capabilities but don't have licenses for specialized tools like HyperLynx, ADS, or AWR.

This article will demonstrate in four steps how to use PSpice to simulate the distortion and common mode rejection of the [Coilcraft 0603USB](#) high speed common mode choke with the manufacturer provided S-parameter model.

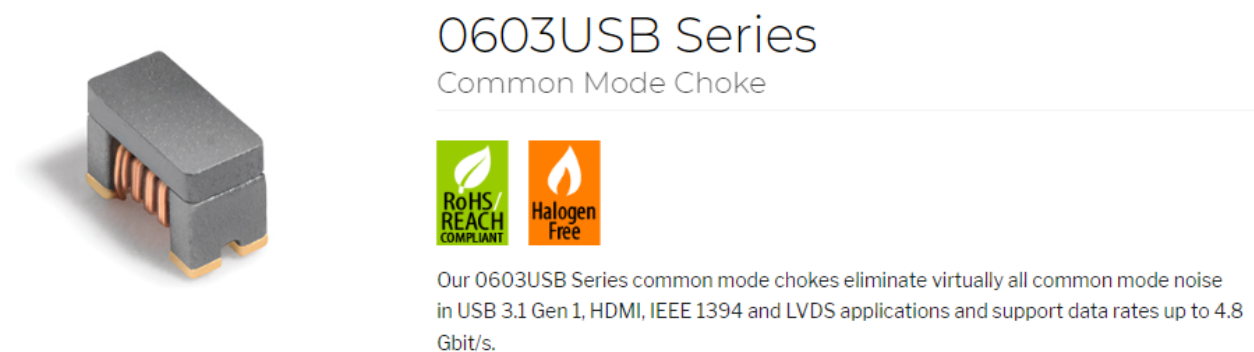


Figure 1. Webpage Listing of Coilcraft 0603USB Common Mode Choke

## Step 1. Gather and Review S-parameter Model

The 0603-222 common mode choke was selected for this analysis with the S-parameter models on the [Coilcraft 0603USB](https://www.coilcraft.com/products/common-mode-chokes/0603-222) webpage. Inspection of the S-parameter file shows:

1. This is a 4 port S-parameter model defined for a 50 Ohm reference.
2. The model is defined from 100 kHz to 8.5 GHz.
3. The model is based on measurements with a VNA, not simulations.
4. The S13/S31 and S24/S42 parameters are ~ 0 dB at low frequencies.

The S-parameter models were also plotted with Python as shown below in Figure 2. Series resonances are seen starting at 2.5 GHz. See the example script below for plotting the S-parameter file.

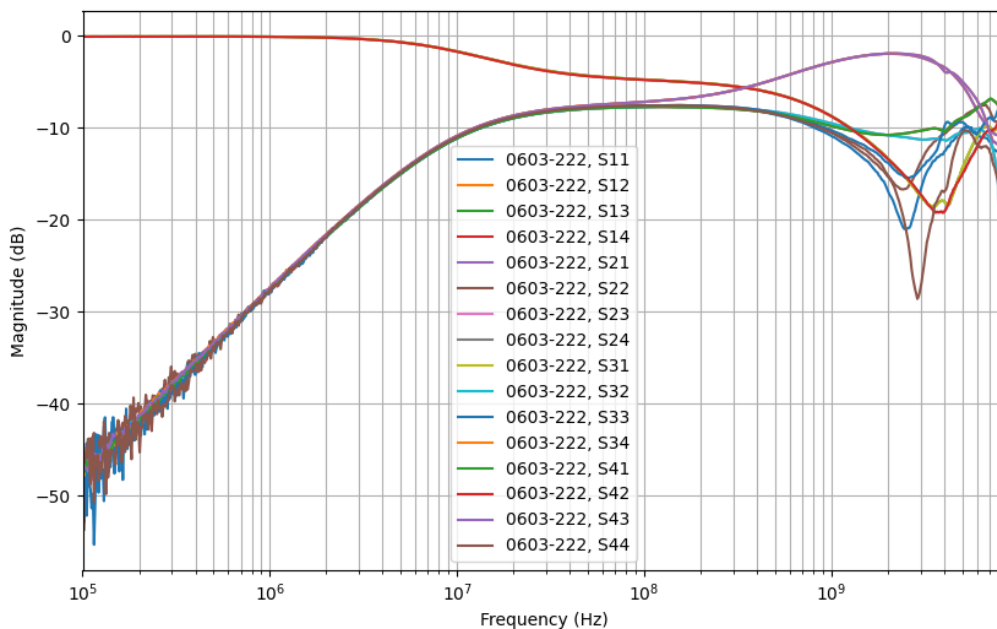


Figure 2. Plot of S-Parameters across Frequency

### Example Python Code for Plotting S-Parameter Files

```
from skrf import Network
import matplotlib.pyplot as plt
ts = Network("0603-222.S4P")
ts.plot_s_db()
plt.gca().set_xscale("log")
plt.grid()
plt.grid(visible=True, which="minor")
plt.show()
```

## Step 2. Convert S-parameter Models to PSpice

The S-parameter models are converted to a PSpice model using the EMI Sleuth [S-parameter to PSpice Converter Tool](#) web page as shown in Figure 3. The PSpice '.lib' library file will automatically be downloaded when conversion is complete. The standard PSpice symbol library can also be downloaded here.

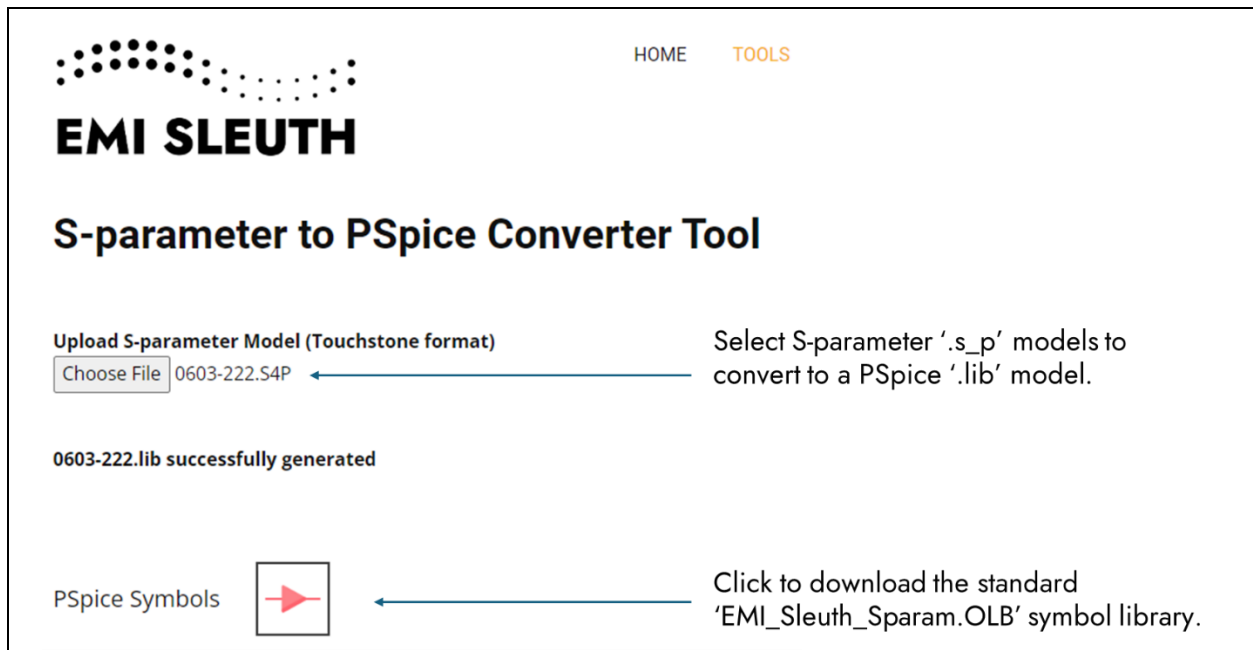


Figure 3. EMI Sleuth S-parameter to PSpice Converter Tool Interface

Once the PSpice '.lib' files are generated they should be reviewed with a text editor. These PSpice library files have a header section with the PSpice model name listed for reference as shown in Table 2.

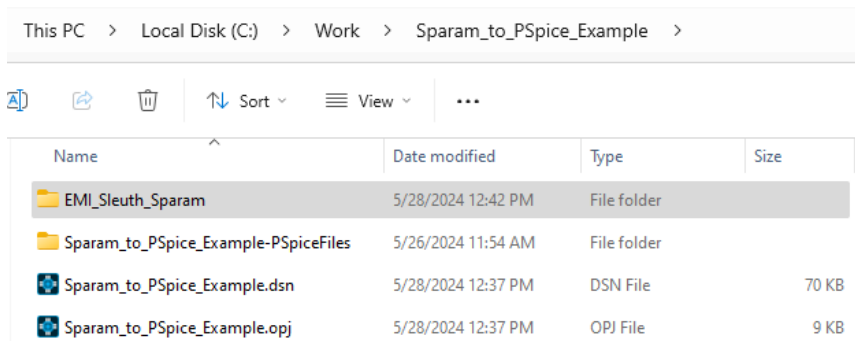
Table 2. PSpice Model Listed in Generated '.lib' Model Header

<b>0603USB Common Mode Choke</b> (0603-222.lib)
* ...
* -- List of PSpice Models --
*
* 0603_222

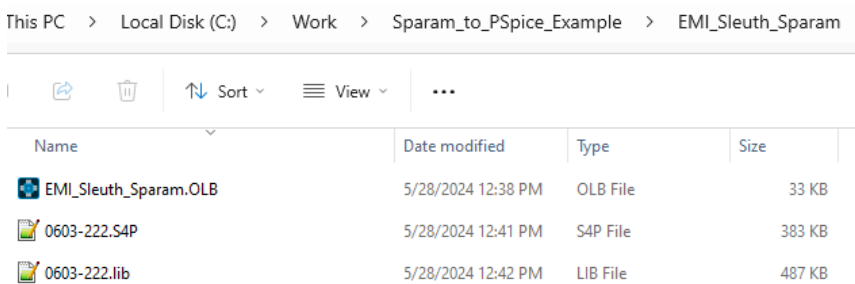
## Step 3. Setup PSpice Project

For this example, a PSpice project was created called "Sparam\_to\_PSpice\_Example". The following steps should be taken to integrate the downloaded PSpice model and symbol libraries:

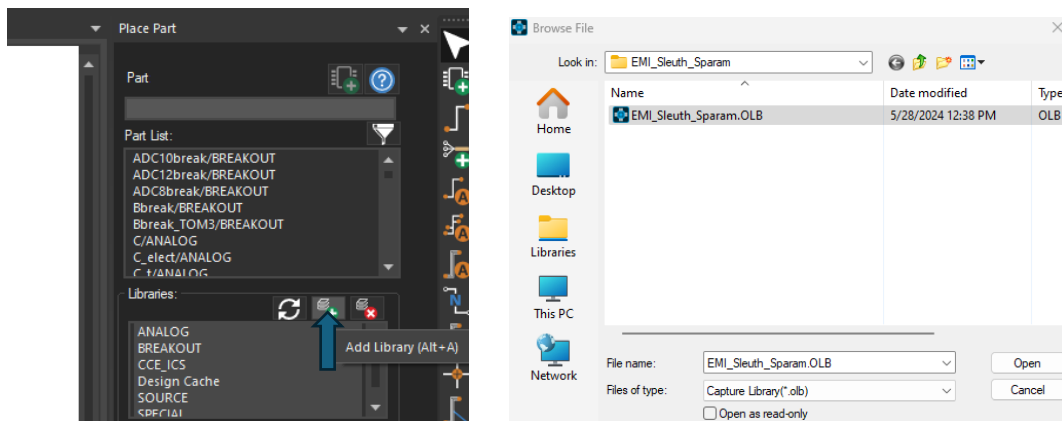
1. Create a new folder inside the project called "EMI\_Sleuth\_Sparam".



2. Add the '.s4p', '.lib' models and the '.OLB' symbol library to the new folder.



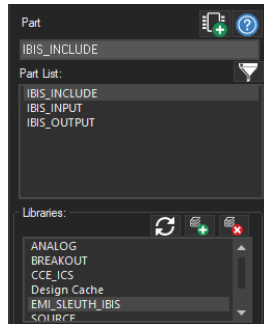
3. Add the "EMI\_Sleuth\_Sparam.OLB" symbol library to the part place libraries.



## Step 4. PSpice Schematic Creation and Simulation

The schematic is created with the following steps:

1. Place the SPARAM\_INCLUDE model into the schematic. Add the names of the converter '.lib' files added in Step 3.

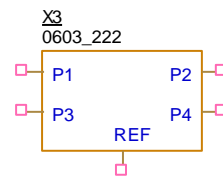
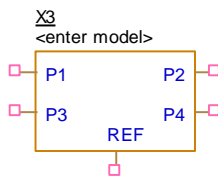
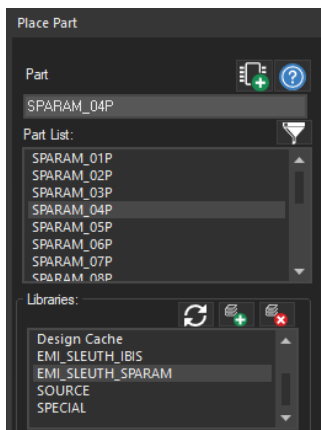


INCLUDE: ../../../../EMI\_Sleuth\_Sparam/<enter library>.lib



INCLUDE: ../../../../EMI\_Sleuth\_Sparam/0603-222.lib

2. Place an SPARAM\_04P 4 port S-parameter symbol into the schematic. Update the model names to match the input and output PSpice models identified in Step 2.



3. Add a driver and a receiver circuits. This simulation will use idealized circuits.
4. For transient simulations, add the SPARAM\_RELTOL symbol. This will update the RELTOL PSpice option and is critical to ensure accuracy at high frequencies. RELTOL is usually set to between 1E-4 and 1E-6 with lower values increasing the simulation time and accuracy. It's usually a good idea to start with RELTOL set high for initial simulations and reduce it for the final simulation.

## How to Use S-parameter Models in PSpice

A completed schematic is shown below in Figure 4. This schematic includes a pair of positive and negative drivers with a 50 Ohm impedance and two 50 Ohm series resistors at the receiver. Voltage probe points are added to driver source, the common mode choke input, and the common mode choke output.

Note that based on the findings of Step 1, ports 1/3 form one winding and ports 2/4 form the second winding. Customizing this symbol to illustrate the windings is recommended to increase readability of the schematic but the generic symbol will be used for this example for simplicity.

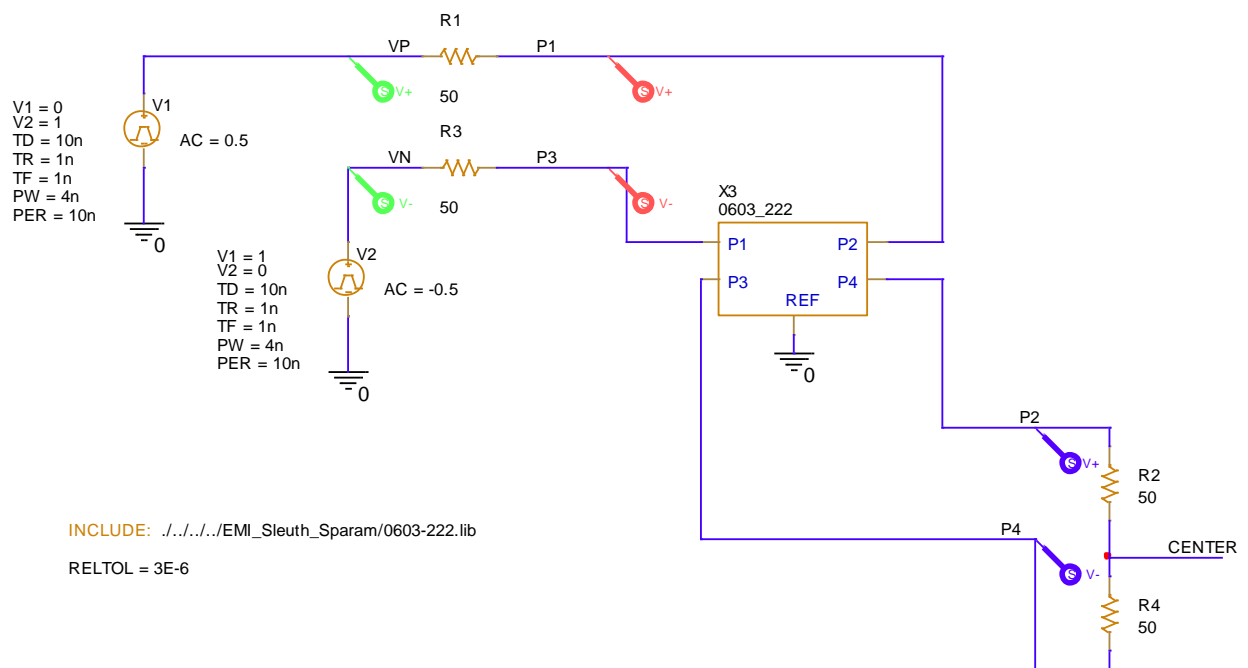


Figure 4. PSpice Schematic of Differential Mode Driver to Common Mode Choke

The simulation in Figure 5 shows little attenuation for a 100 MHz differential driver, though attenuation starts to increase above 300 MHz and may be significant for those higher speed interfaces. The time domain simulation shows a slight attenuation and a 0.2 ns propagation delay through the common mode choke.

# How to Use S-parameter Models in PSpice

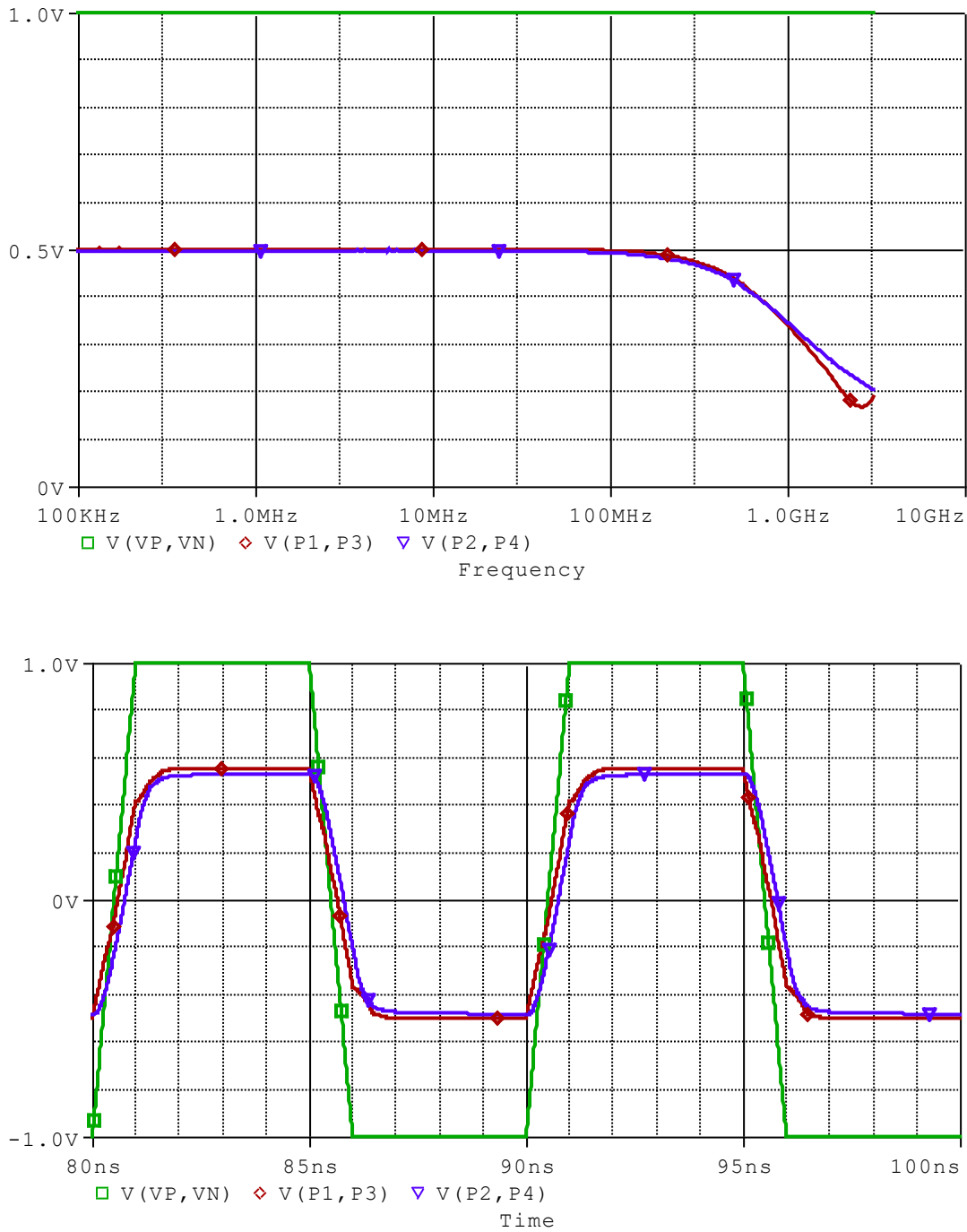


Figure 5. Frequency (Top) and Time (Bottom) Domain Simulations of Common Mode Choke with Differential Mode Drivers

## How to Use S-parameter Models in PSpice

The circuit was modified to simulate common mode rejection of the choke as shown below in Figure 6. Negligible common mode rejection is seen in Figure 7, and common mode noise is increased above 2 GHz due to resonances of the choke.

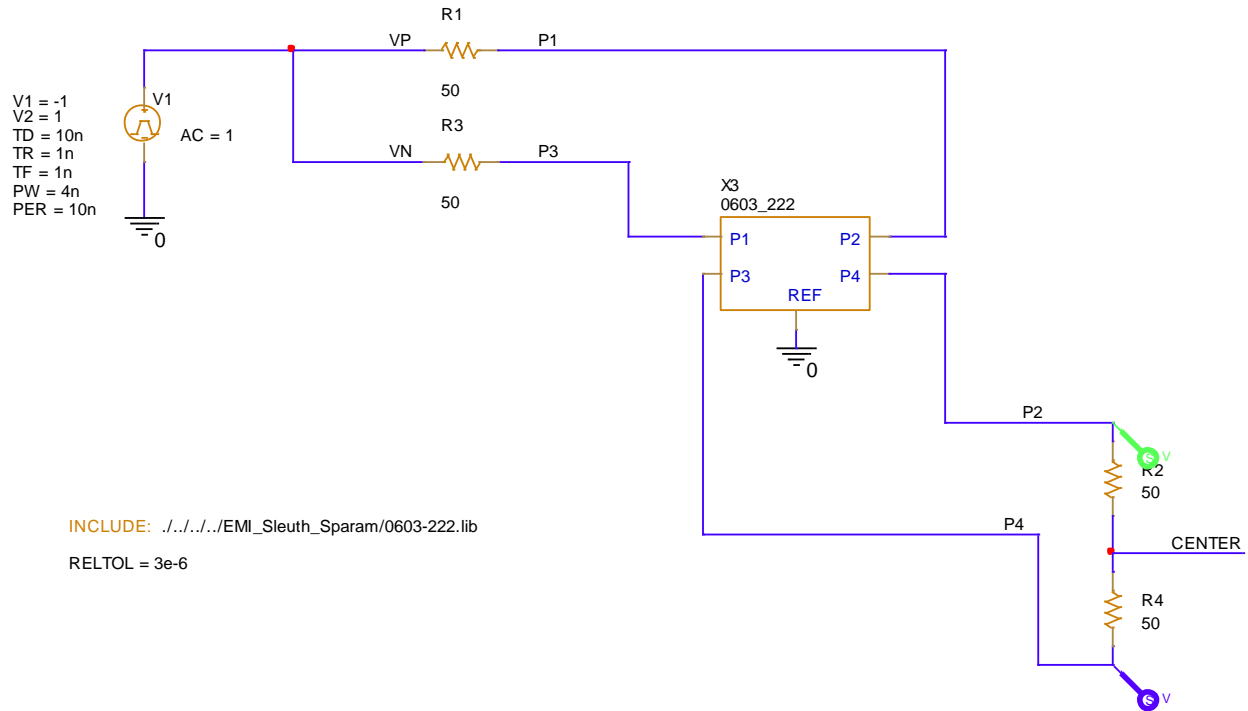


Figure 6. PSpice Schematic of Common Mode Rejection with No Center Tap Capacitance

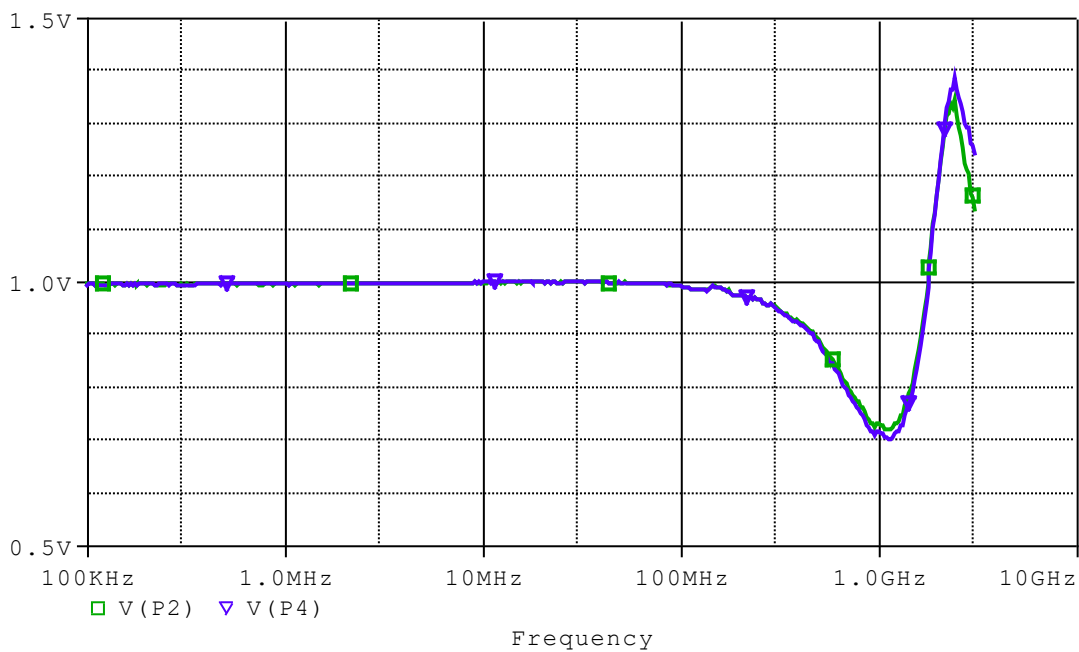


Figure 7. Common Mode Coupling with No Center Tap Capacitance





### Conclusions

This article demonstrated that using accurate S-parameter models in PSpice is possible with the EMI Sleuth [S-parameter to PSpice Converter Tool](#). A PSpice simulation of a common mode choke with the manufacturer provided 4 port S-parameter model was performed in the time domain and frequency domains. A differential mode analysis showed minimal distortion of the 100 MHz signal due to the choke. The same circuit was analyzed for common mode rejection, but negligible filtering was observed with the original circuit. A capacitive load to the center of the receiver termination resistors was added and the common mode rejection was seen to significantly improve.

Therefore, just adding a common mode choke was not enough to improve common mode rejection at the differential receiver. Adding capacitive loading was also required, and selection of the capacitance required a PSpice simulation using an accurate S-parameter model of the common mode choke.