How to Anticipate and Mitigate Conducted Emissions for a Buck Converter Using LTspice

Table Of Contents

Table Of Contents

Introduction

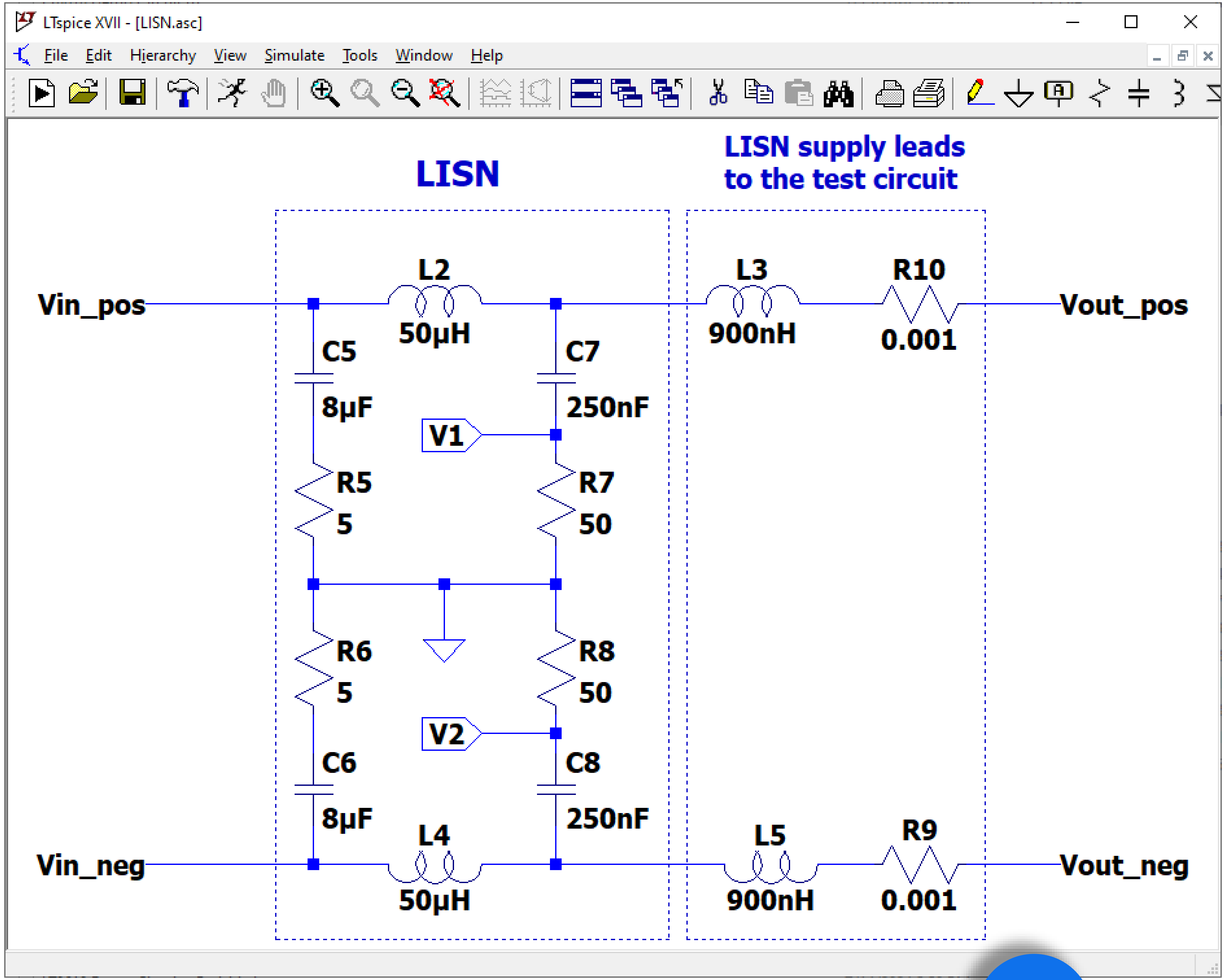
Buck converters are integral to modern electronics, providing efficient DC-DC conversion across a wide range of applications, from consumer gadgets to industrial equipment. ***Despite their efficiency and utility, these converters pose a significant challenge: conducted emissions***. The high-frequency switching necessary for their operation generates noise that can propagate through power lines, leading to electromagnetic interference (EMI). This interference can degrade the performance of the buck converter itself and disrupt nearby electronic devices. Ensuring that these emissions are managed and mitigated is crucial for both regulatory compliance and the reliable operation of electronic systems.

LTspice, a robust circuit simulation tool, offers engineers a powerful means to tackle this challenge. By using LTspice to simulate conducted emissions, engineers can identify potential noise issues early in the design process and develop effective strategies to mitigate them. This proactive approach not only saves time and resources but also ensures that the final product adheres to EMI regulations and

buck converter itself and disrupt nearby electronic devices. Ensuring that these emissions are managed and mitigated is crucial for both regulatory compliance and the reliable operation of electronic systems.

LTspice, a robust circuit simulation tool, offers engineers a powerful means to tackle this challenge. By using LTspice to simulate conducted emissions, engineers can identify potential noise issues early in the design process and develop effective strategies to mitigate them. This proactive approach not only saves time and resources but also ensures that the final product adheres to EMI regulations and performs reliably in real-world scenarios. In this article, we will explore how to anticipate and mitigate conducted emissions in buck converters using LTspice, covering the fundamentals of conducted emissions, the setup of accurate simulations, and practical mitigation techniques to enhance the EMI performance of your designs.

LTspice LISN Circuit Used To Measure Common Mode (CM) Noise

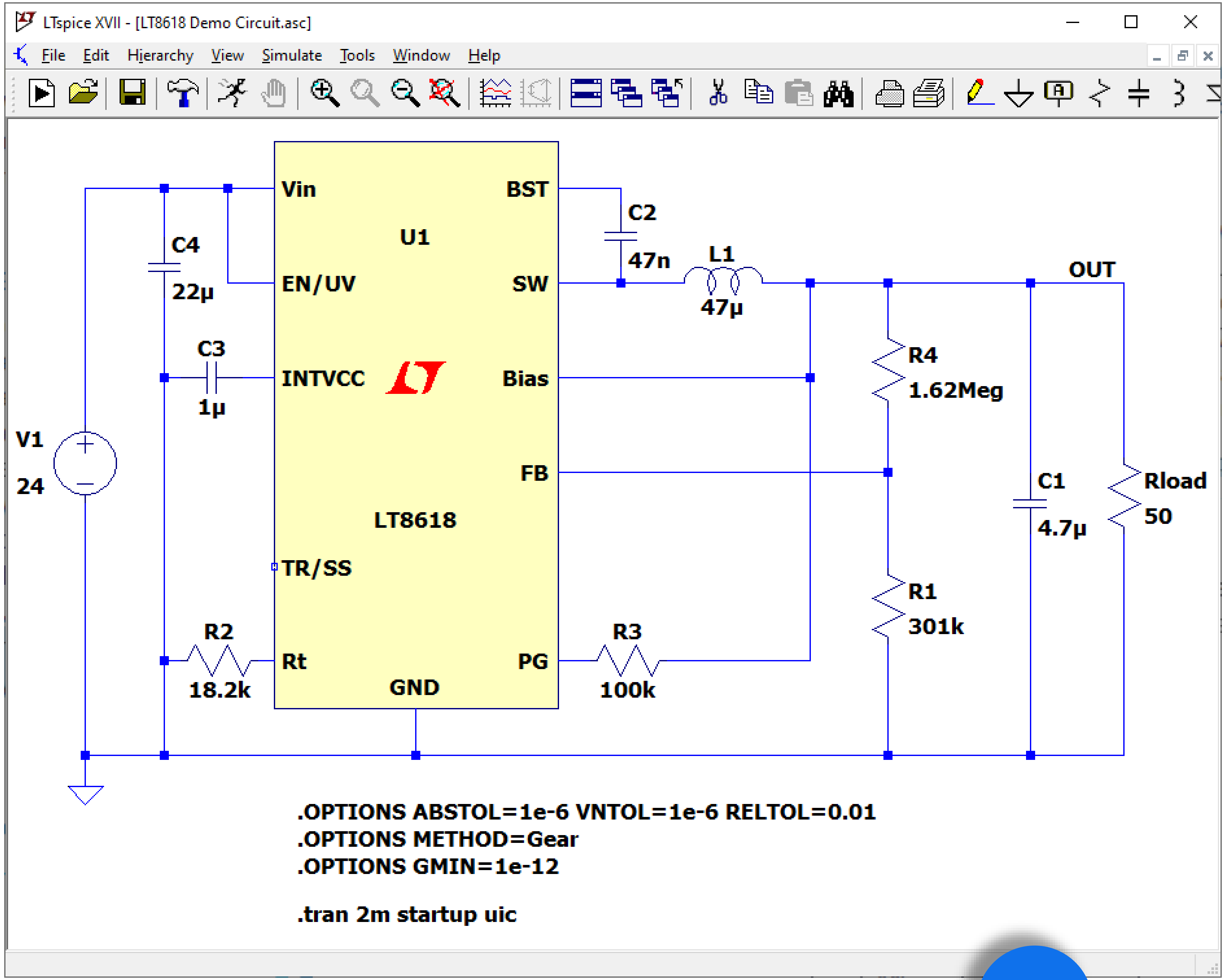


This circuit is the LTspice **LISN** circuit used in this article.

A Line Impedance Stabilization Network (**LISN**) is a test fixture used in EMC (Electromagnetic Compatibility) testing to provide a stable and repeatable impedance to the device under test (DUT) and to measure the conducted emissions on the power lines.

When simulating conducted emissions for a buck converter in LTspice, incorporating a LISN model can help predict the emissions and evaluate mitigation strategies.

Circuit Under Test Schematic



Circuit Description:

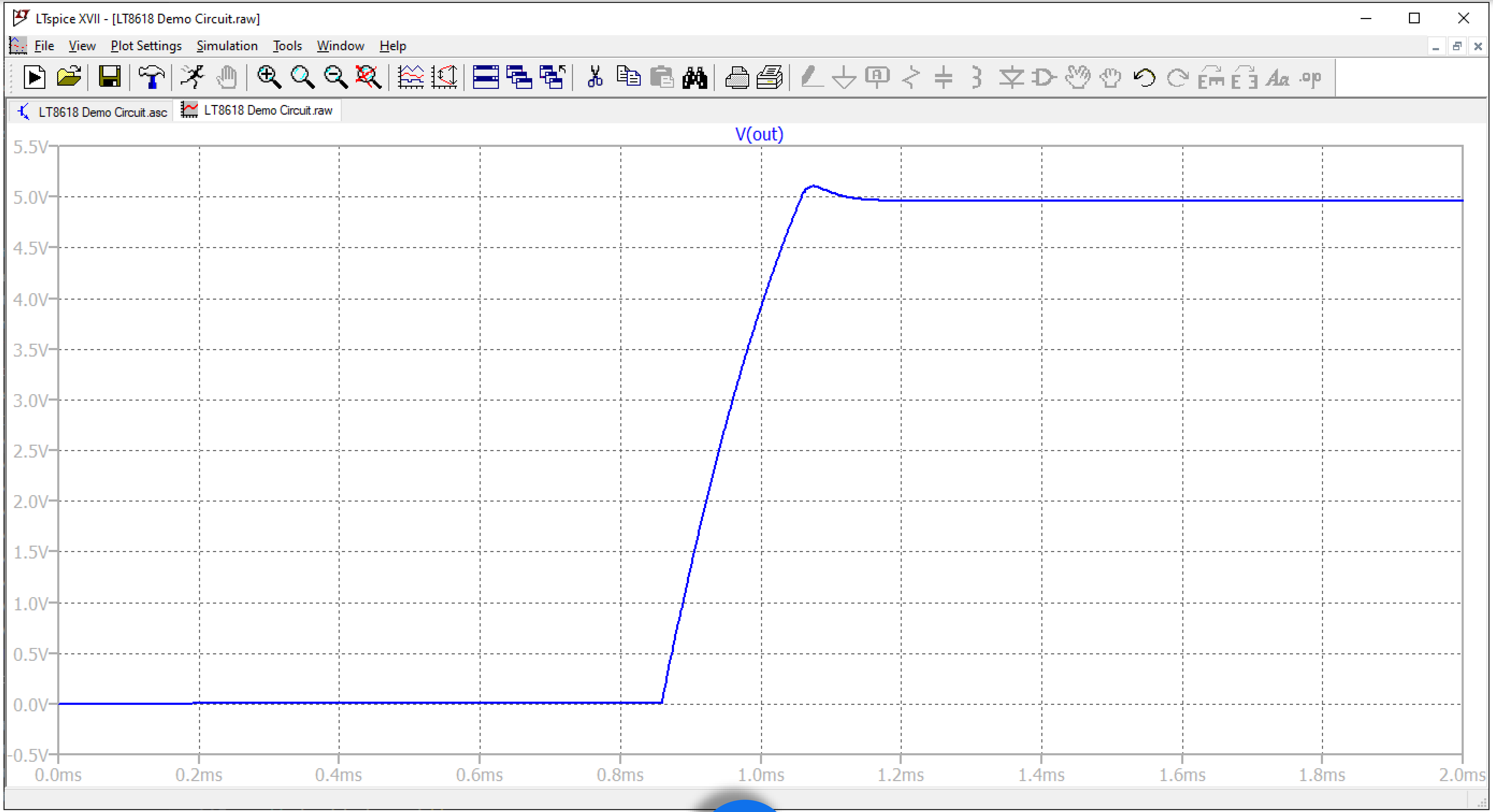
This circuit is a step-down converter based on LT8618 chip.

Input: 24V

Output: 5V @ 100mA

Fsw: 2MHz

Circuit Under Test Transient Response



Simulation Parameters

To improve simulation speed in LTspice several simulation parameters were used. The following is the explanation of the parameters:

Parameter: ABSTOL=1e-6

Function: This sets the absolute tolerance for the current in the circuit. It defines the smallest current difference that the simulator will consider as converged.

Impact: A smaller value increases accuracy but can slow down the simulation because the simulator will take smaller steps to meet the stricter tolerance.

Parameter: VNTOL=1e-6

Function: This sets the absolute tolerance for the voltage in the circuit. It specifies the smallest voltage difference that the simulator will treat as converged.

Impact: Similar to ABSTOL, a smaller value enhances accuracy but can result in slower simulation times due to the need for more refined calculations.

Simulation Parameters

Parameter: RELTOL=0.01

Function: This sets the relative tolerance for both voltage and current. It determines the acceptable relative error as a percentage of the value being calculated.

Impact: A smaller relative tolerance (e.g., 0.01, or 1%) means stricter convergence criteria, leading to more accurate results but potentially slower simulations.

Parameter: METHOD=Gear

Function: This option in LTspice specifies that the Gear method should be used for numerical integration during transient analysis.This method uses multiple past points to estimate the current point, enhancing accuracy and stability over single-step methods.

Impact: The Gear method is particularly advantageous for stiff systems, where there are components with widely varying time constants. Examples include power electronics with rapid switching transients and slower reactive components like large inductors and capacitors.

Simulation Parameters

Function: This option in LTspice specifies that the Gear method should be used for numerical integration during transient analysis.This method uses multiple past points to estimate the current point, enhancing accuracy and stability over single-step methods.

Impact: The Gear method is particularly advantageous for stiff systems, where there are components with widely varying time constants. Examples include power electronics with rapid switching transients and slower reactive components like large inductors and capacitors.

Parameter: GMIN=1e-12

Function: The primary function of GMIN is to help the simulator achieve convergence. Numerical instabilities can occur in circuits with high resistance or very small currents. The GMIN parameter helps to mitigate these instabilities by ensuring that every node has a defined, albeit very small, path to ground.

Impact: Setting GMIN=1e-12 can significantly improve the convergence of simulations. It prevents the simulator from struggling with floating nodes.

Running a Fast Fourier Transform (FFT) in LTspice

Performing a Fast Fourier Transform (**FFT**) in LTspice allows you to analyze the frequency content of a signal, which is particularly useful for examining harmonics, noise, and other spectral characteristics in your circuits. In our case the FFT will be used to analyze the Common Mode (CM) Noise.

Here's a step-by-step guide on how to perform an FFT in LTspice:

Step 1: Set Up Your Circuit

Ensure that the circuit is properly configured and that you have defined a transient analysis to capture the signal you want to analyze.

Step 2: Run the Transient Simulation

Run a transient analysis by including the .TRAN directive in your schematic. This analysis will generate the time-domain data needed for the FFT.

Step 3: View the Time-Domain Waveform

Running a Fast Fourier Transform (FFT) in LTspice

Step 3: View the Time-Domain Waveform

After running the simulation, view the time-domain waveform by clicking on the nodes of interest. In the circuits shown in the article, you would click on **V1** and **V2** to view the voltage across the capacitor.

Step 4: Perform the FFT

Open the FFT Window: Right-click on the waveform viewer and select "View" -> "FFT" from the context menu.

Step 5. Select Waveforms to include in FFT

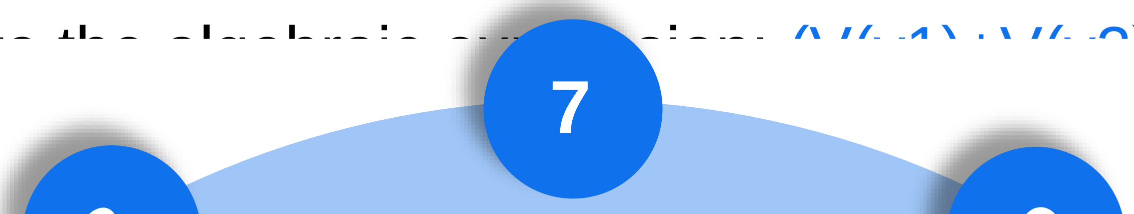
In the FFT window, select the signals to analyze. Ensure you have the correct waveforms selected (V(v1) and V(v2)).

Step 6: Select Visible Waveforms

In the windows select both signals V(v1) and V(v2)

Step 7: Create the Algebraic Expression for Common Mode (CM) Noise

Use the expression editor and create the algebraic expression: (V(v1)+V(v2))*0.5*1000000



Running a Fast Fourier Transform (FFT) in LTspice

Step 4: Perform the FFT

Open the FFT Window: Right-click on the waveform viewer and select "View" -> "FFT" from the context menu.

Step 5. Select Waveforms to include in FFT

In the FFT window, select the signals to analyze. Ensure you have the correct waveforms selected (V(v1) and V(v2)).

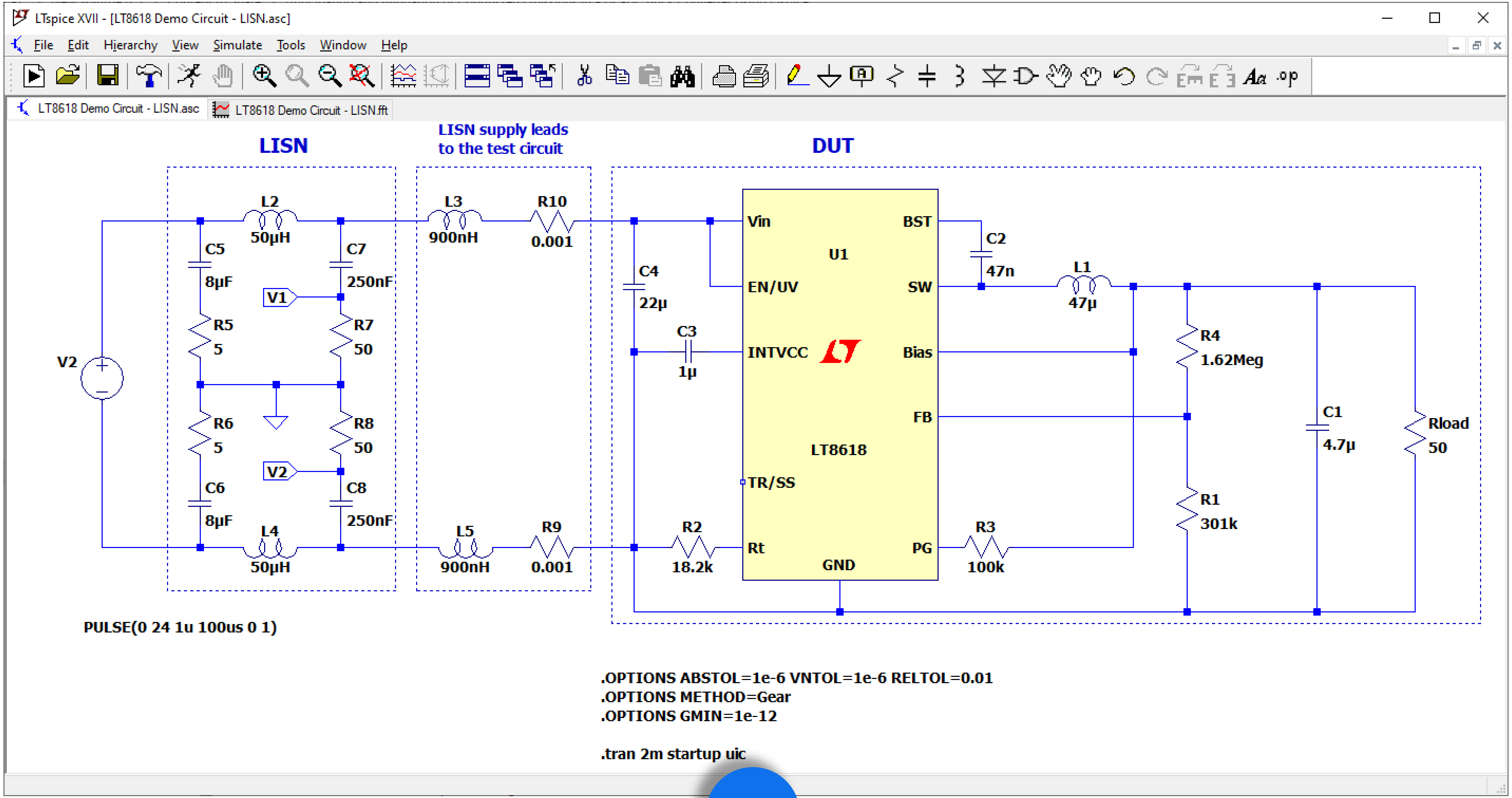
Step 6: Select Visible Waveforms

In the windows select both signals V(v1) and V(v2)

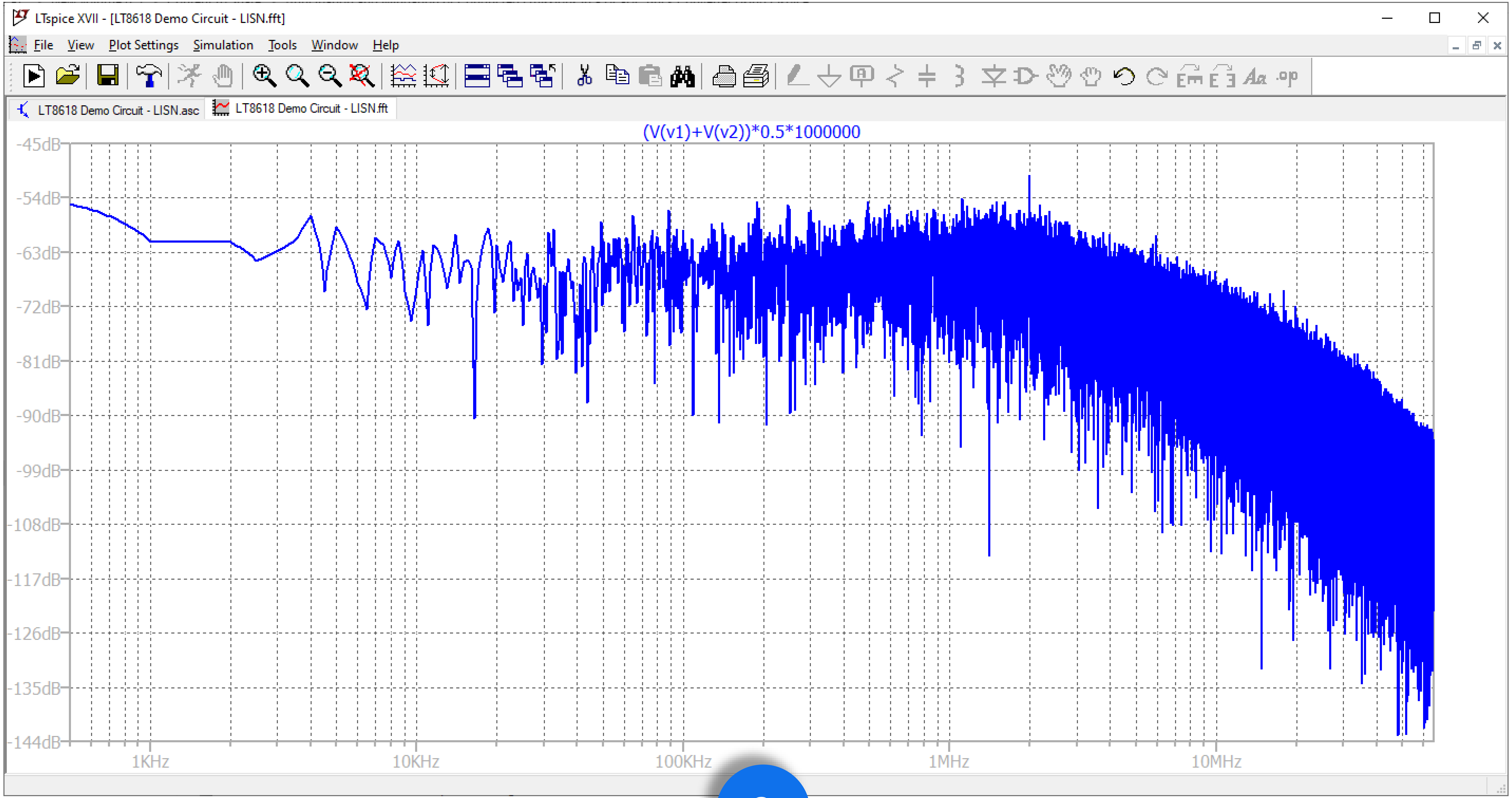
Step 7: Create the Algebraic Expression for Common Mode (CM) Noise

Use the expression editor and create the algebraic expression: (V(v1)+V(v2))*0.5*1000000

Adding a LISN to Circuit Under Test

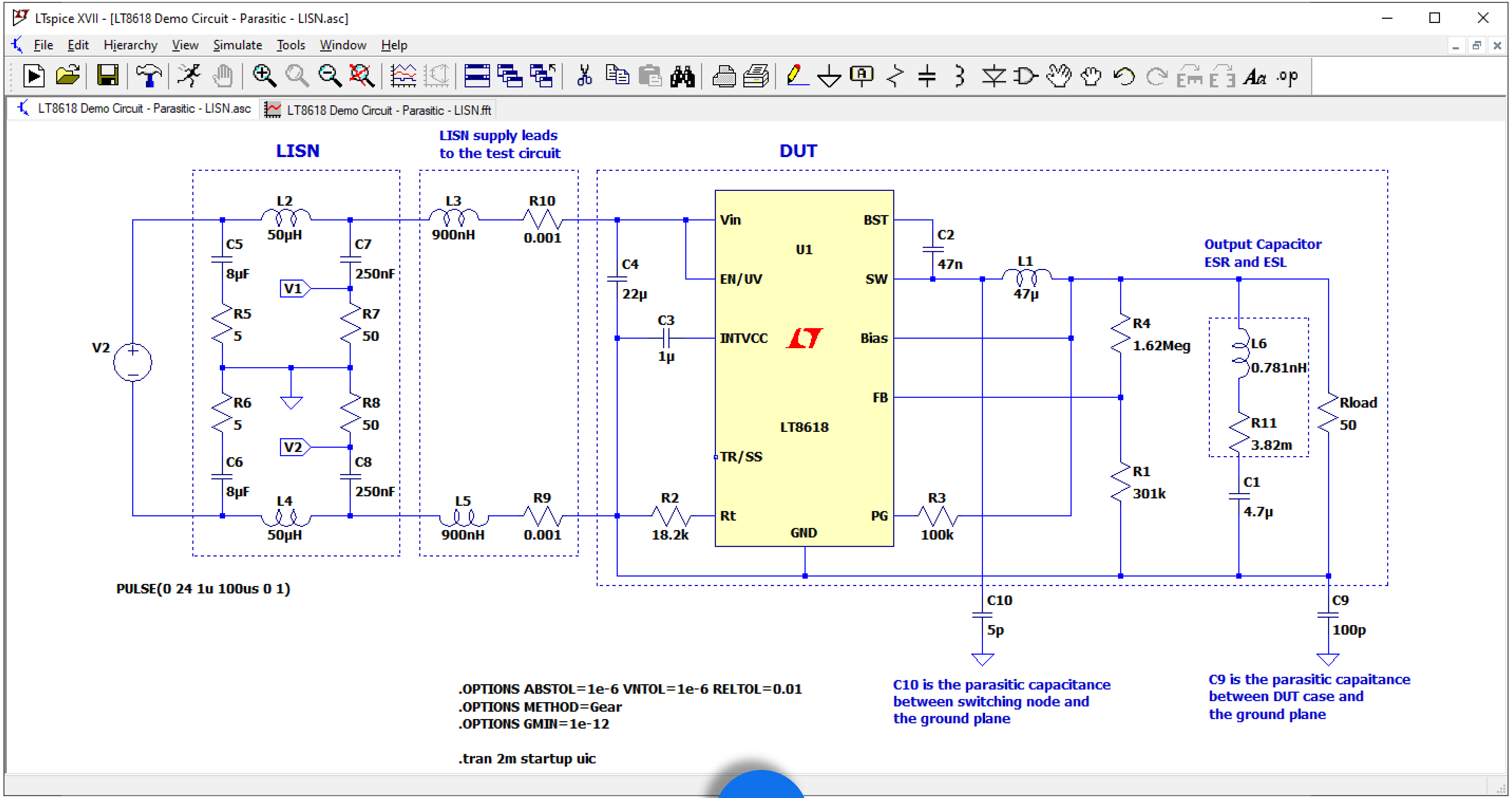


Common Mode (CM) Noise Response

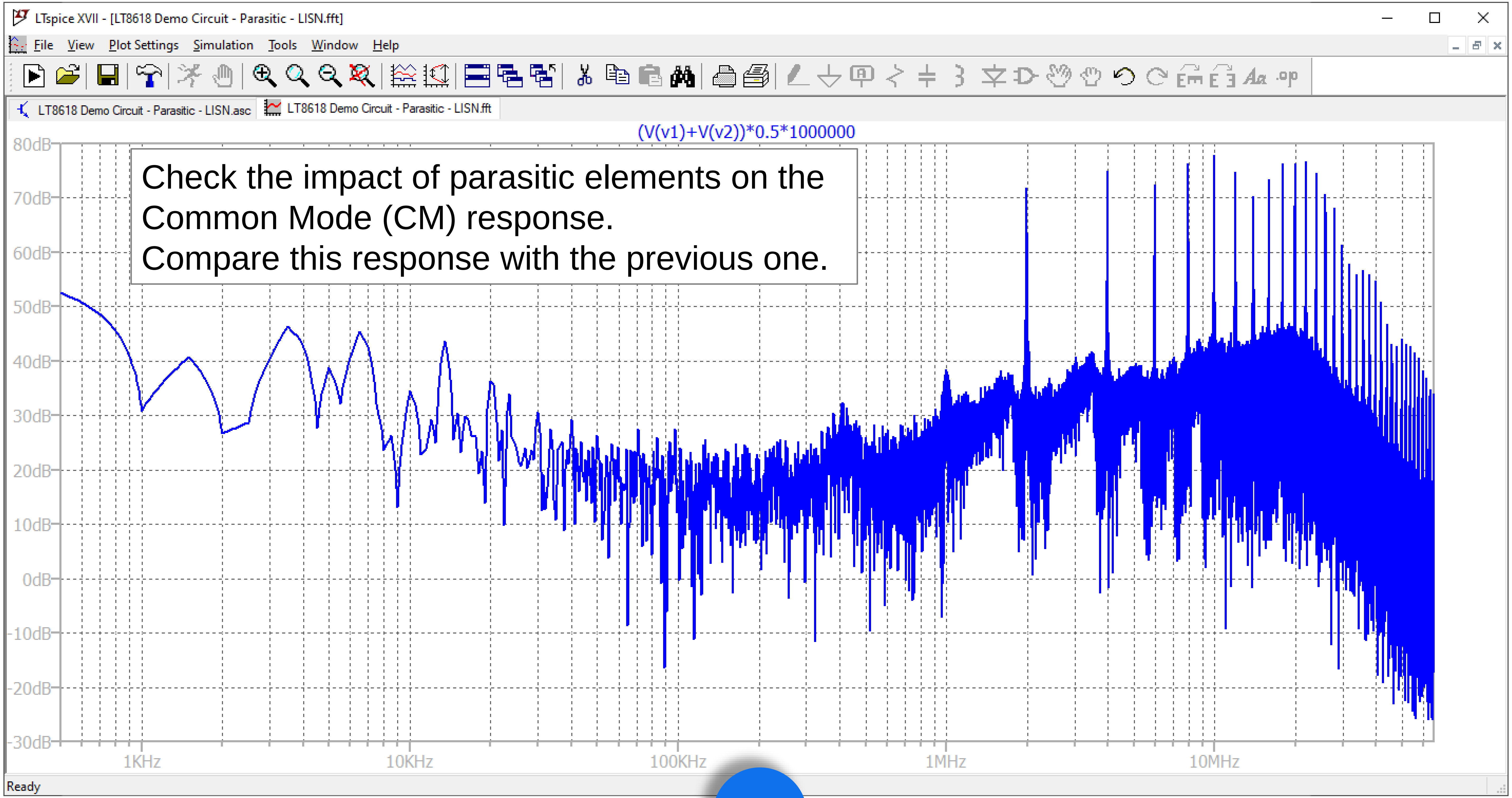


9

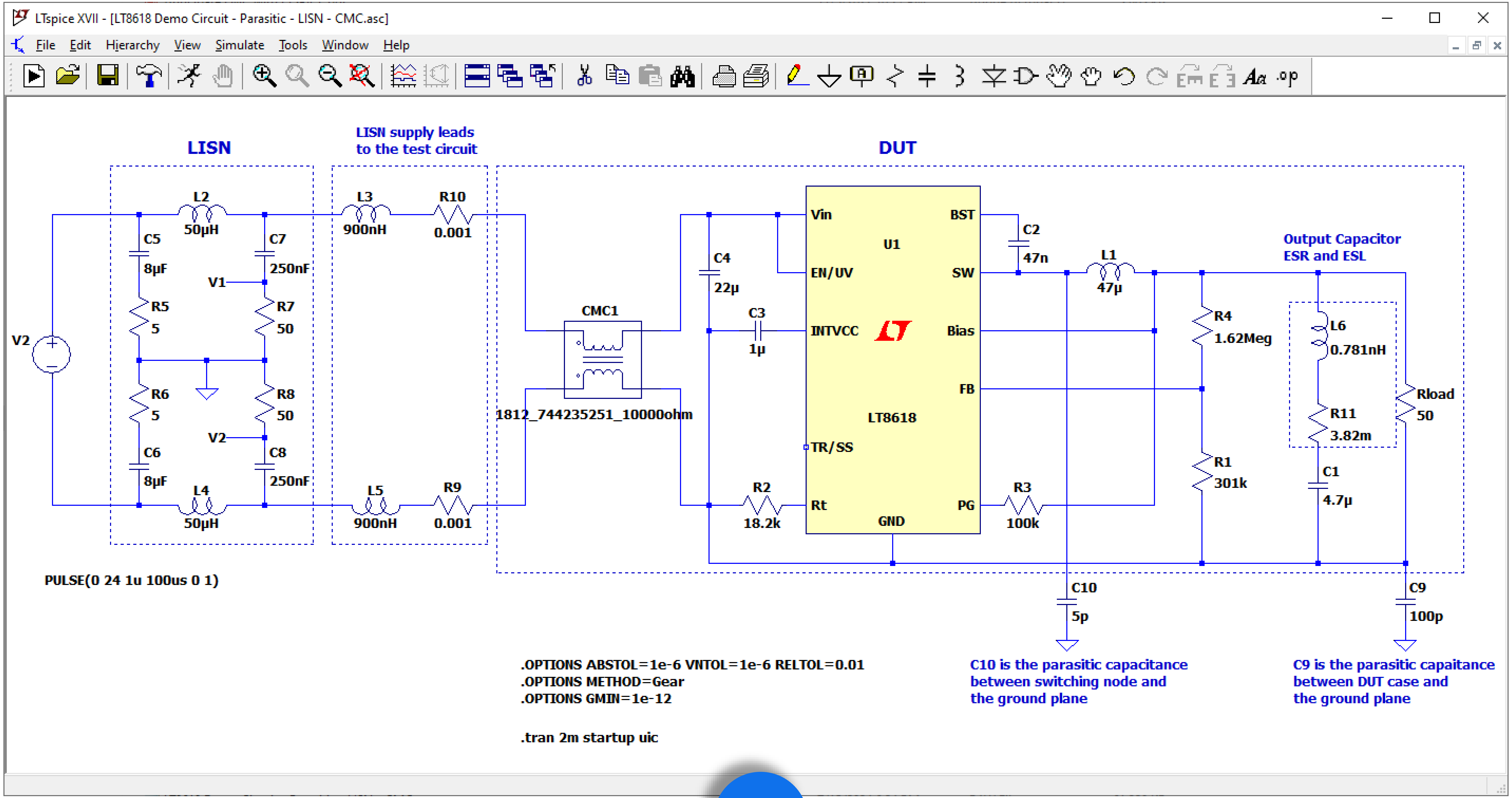
Adding Parasitic Elements to Circuit Under Test



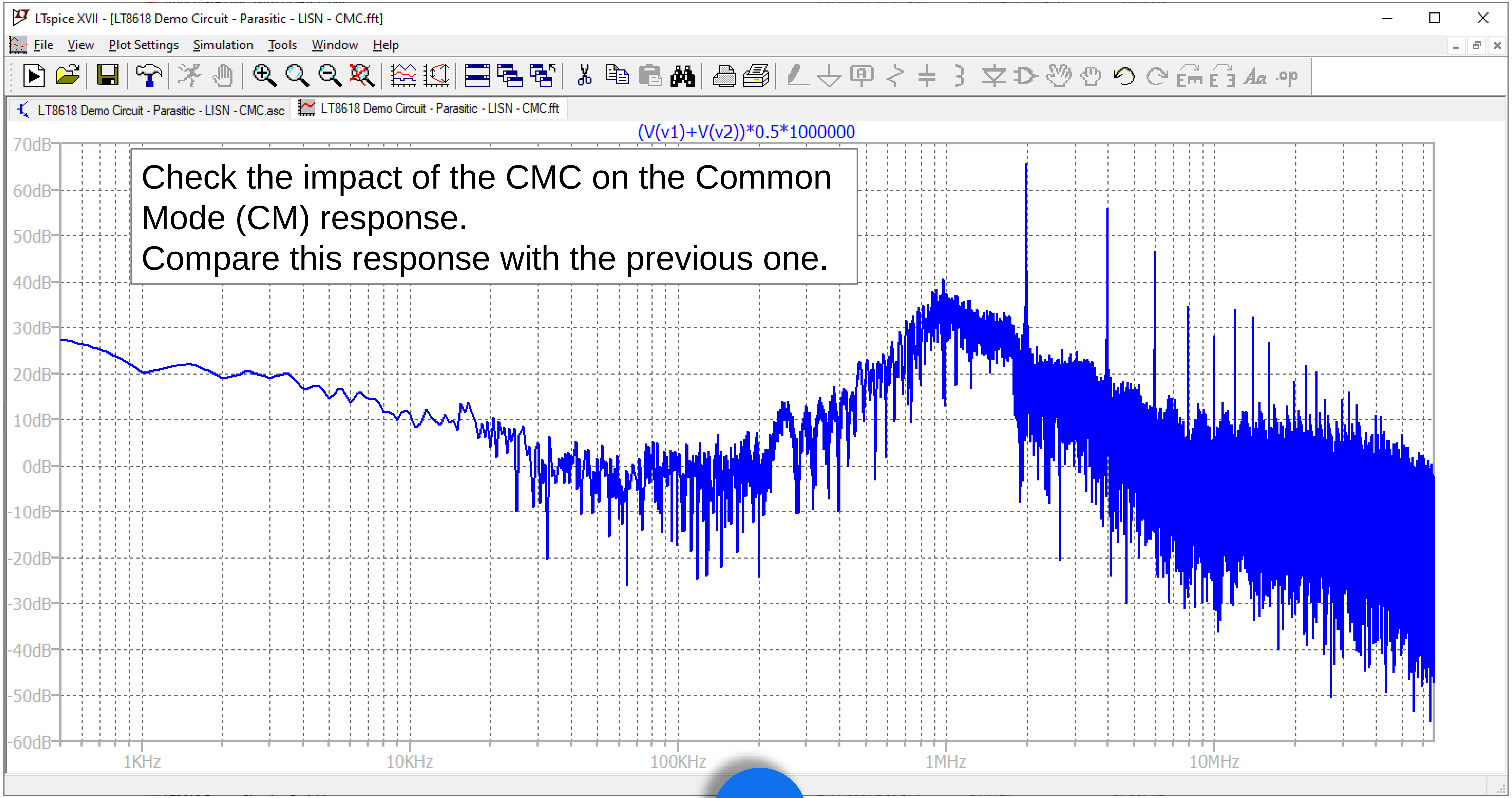
Common Mode (CM) Noise Response when Parasitic Elements are Present



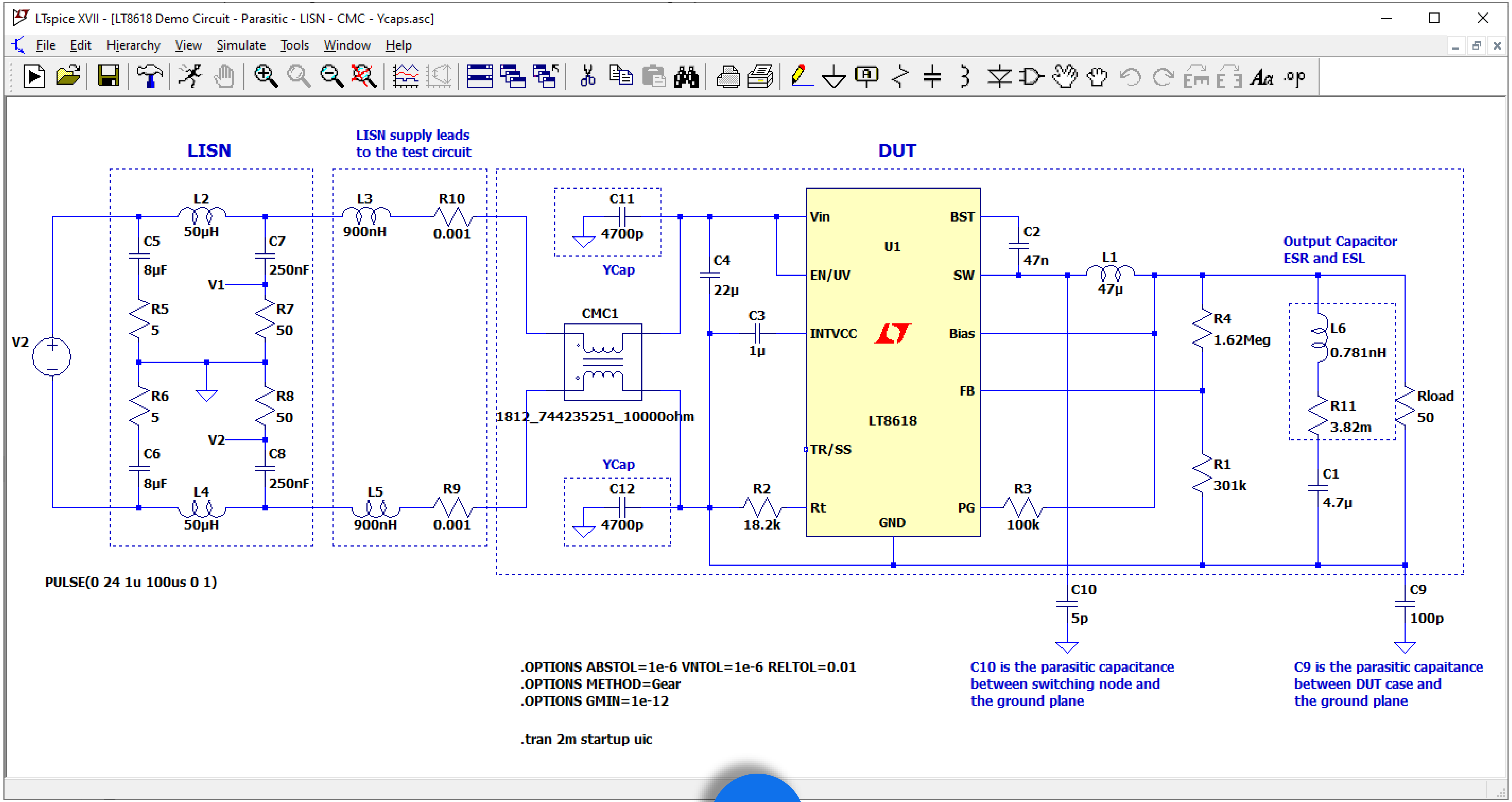
Adding a CMC Between the LISN and Circuit Under Test



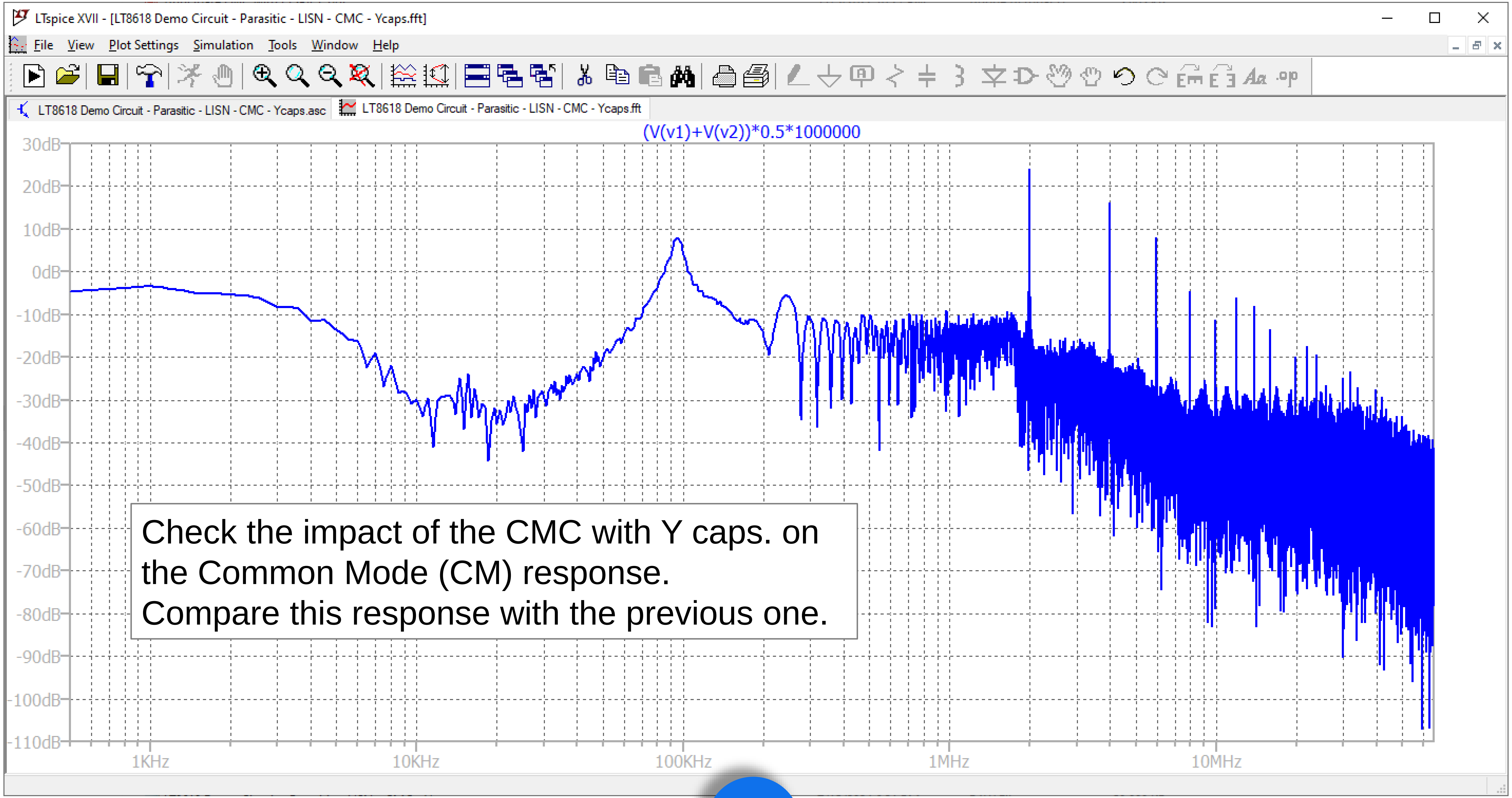
Common Mode (CM) Noise Response when CMC is Present



Adding Y Capacitors to CMC



Common Mode (CM) Noise Response when Y Capacitors are Present



Check the impact of the CMC with Y caps. on the Common Mode (CM) response.
Compare this response with the previous one.

Conclusion

Adding a common mode choke (CMC) to the input of a buck converter is an effective strategy for mitigating common mode noise and achieving EMI (Electromagnetic Interference) compliance.

The CMC presents high impedance to common mode currents, which are currents that flow in the same direction on both conductors of a pair (e.g., both the positive and negative input lines of a buck converter).

The CMC coils are magnetically coupled, which enhances the choke's ability to reject common mode noise without significantly affecting the desired power signal.

The CMC should be placed as close as possible to the input of the buck converter to maximize its effectiveness in blocking noise before it propagates into the circuit.

Choosing a CMC with the appropriate impedance characteristics is crucial. The choke should provide sufficient attenuation at the frequencies of interest while not significantly affecting the differential mode signal.

"Thanks for watching! If you enjoyed this video, make sure to hit the like button and subscribe to stay updated with my latest content. Don't forget to check out my other videos for more tips and tutorials on Embedded C, Python, hardware designs, etc. Keep exploring, keep learning, and I'll see you in the next video!"