Signal chain LTspice simulation guide

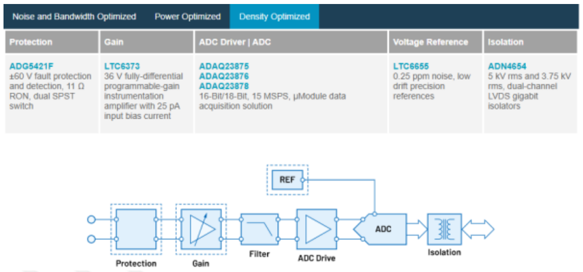
Precision wide-bandwidth current or voltage measurement, density optimized

Introduction

This guide offers a walkthrough on simulating a precision wide bandwidth measurement signal chain optimized for density, using the accompanying LTspice schematic (.asc file). AC, transient and noise simulation examples are shown. A basic level of familiarity with LTspice is assumed, those new to the tool might want to go over a primer such as [1] first and make sure to check the attached readme.txt file. For support, you can post your questions in the LTspice forum in ADI Engineer Zone [2]. You will need to create an account and log in to post questions.

Signal chain presentation

A block diagram of the signal chain can be seen in Figure 1.



Protection	Gain	ADC Driver | ADC	Voltage Reference	Isolation
ADG5421F ±60 V fault protection and detection, 11 Ω RON, dual SPST switch	**LTC6373** 36 V fully-differential programmable-gain instrumentation amplifier with 25 pA input bias current	**ADAQ23875** **ADAQ23876** **ADAQ23878** 16-Bit/18-Bit, 15 MSPS, µModule data acquisition solution	**LTC6655** 0.25 ppm noise, low drift precision references	**ADN4654** 5 kV rms and 3.75 kV rms, dual-channel LVDS gigabit isolators

Figure 1.- Signal chain block diagram

Additional information in relation to this signal chain is available in Analog Device's website [3].

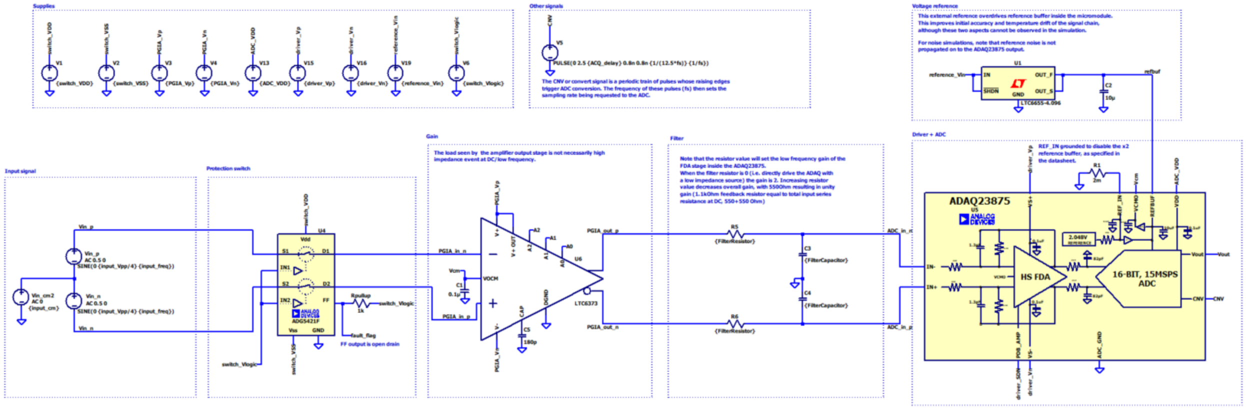
For this implementation of the signal chain, we have chosen the ADAQ23875 µModule® which has a 16-bit ADC. The ADAQ23876 is an alternative that could provide additional options in terms of gain and input range, while the ADAQ23878 would offer those choices and also increase the ADC resolution to 18-bit.

We are using an external refence, a 4.096 V LT6655, overdriving the internal one in the ADAQ23875. This external reference has better overall specs, most notably a temperature drift one order of magnitude below that of the internal one (2 ppm/°C vs 20 ppm/°C).

For the signal filter we will be using a first order differential RC network. The input stage of the ADAQ23875 is not high impedance, which means that the resistor value that we choose for the filter will affect signal chain gain. We will set the resistors (R5, R6 in Figure 2 schematic) to 550 Ω, which will cause these stages to have a gain of 1. This means that the overall gain of the signal chain gain will be set by the PGIA at the front alone. As for its 3 dB bandwidth, we will assume that a value of 100 kHz is adequate for the input signal needs of the application at hand. This is achieved with a filter capacitor value of 5.8 nF. For how the resistor values affect signal chain gain, and how to calculate the capacitor value, see the section *FDA gain and signal filter bandwidth* below at the end of this guide. Additional information and other options for signal chain filtering can be found here [4].

Signal chain LTspice simulation guide

Precision wide-bandwidth current or voltage measurement, density optimized



The CNV or convert signal is a periodic train of pulses whose raising edges trigger ADC conversion. The frequency of these pulses (fs) then sets the sampling rate being requested to the ADC.

This external reference overdrives reference buffer inside the micromodule. This improves initial accuracy and temperature drift of the signal chain, although these two aspects cannot be observed in the simulation.

For noise simulations, note that reference noise is not propagated on to the ADAQ23875 output.

FF output is open drain

The load seen by the amplifier output stage is not necessarily high impedance event at DC/low frequency.

Note that the resistor value will set the low frequency gain of the FDA stage inside the ADAQ23875. When the filter resistor is 0 (i.e. directly drive the ADAQ with a low impedance source) the gain is 2. Increasing resistor value decreases overall gain, with 550Ohm resulting in unity gain (1.1kOhm feedback resistor equal to total input series resistance at DC, 550+550 Ohm)

REF_IN grounded to disable the x2 reference buffer, as specified in the datasheet.

Figure 2.- Signal chain LTspice schematic

Simulation file usage

The accompanying LTspice simulation file is divided in boxed sections for readability. Some of these sections are:

Simulation commands

LTspice simulation commands for three kinds of simulation are located in this block, namely, transient, AC and noise. LTspice requires that only one simulation command exists in the schematic, so the most convenient use of this block is to comment out all commands except for the one we intend to run. Turning a simulation command into a comment can be quickly done by right clicking on the text box containing it, pressing escape to dismiss the simulation dialog, and then, in the newly appearing one, choosing "Comment" on the "How to netlist this text" control. The reverse operation is quicker, as right clicking on a comment will show the dialog where we can select "SPICE directive" as the way we want to netlist the text.

Parameters

Most variable elements in this signal chain can be controlled by modifying the parameters defined in this section. LTspice parameters are created through the .PARAM command. This allows quick modification of the signal chain, keeping track of its current status and it is convenient for running stepped simulations (if unfamiliar with .STEP command, see [5]).

The gain of the LT6373 PGIA used in this signal chain is set through 3 digital inputs. However, in this simulation it can be controlled through the single *PGIA_gain* parameter, as there is a section that decodes this parameter into the corresponding combination of digital outputs ("LT6373 gain decode" auxiliary block).

Other signals

The generation of the "CNV" (convert) signal that triggers the ADC can be found here. This is a pulse train whose rising edges trigger the conversion process. See this reference [6] in relation to waveform generation in LTspice.

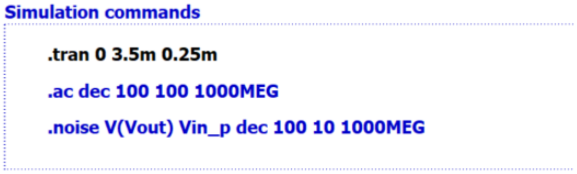
Transient simulation

Transient simulation allows us to observe the signals in our circuit in the time domain; it is done through the .TRAN command.

Transient simulation example: verifying circuit gain

In this example we will use a transient simulation to check that the gain set in the simulation parameters for the amplifier stage is applied properly to an input signal.

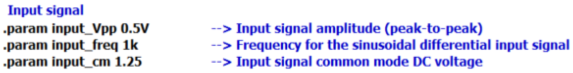
We start out by commenting out all simulation commands except for the .TRAN one, after which our simulation command box should look like this:



```
Simulation commands

    .tran 0 3.5m 0.25m

    .ac dec 100 100 1000MEG

    .noise V(Vout) Vin_p dec 100 10 1000MEG
```

Figure 3.- TRAN example: simulation commands

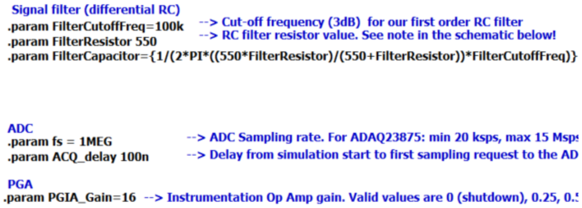
For our input signal, we will setup a 500 mV peak-to-peak, differential, 1 kHz sinusoidal signal with 1.25V of common mode, through these parameters:



```
 Input signal
.param input_Vpp 0.5V           --> Input signal amplitude (peak-to-peak)
.param input_freq 1k            --> Frequency for the sinusoidal differential input signal
.param input_cm 1.25            --> Input signal common mode DC voltage
```

Figure 4.- TRAN example: input signal configuration

As for the signal chain configuration, we set the PGIA gain to 16, expecting then to have an 8 Vpp signal at the output of the signal chain and make use of practically the whole input range of the ADC (±4.096 V). Our previous election of a 100 kHz bandwidth for the RC filter leaves the 1 kHz input well within the passband region.

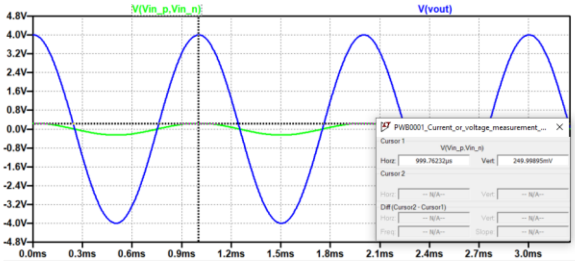


```
 Signal filter (differential RC)
.param FilterCutoffFreq=100k   --> Cut-off frequency (3dB)  for our first order RC filter
.param FilterResistor 550      --> RC filter resistor value. See note in the schematic below!
.param FilterCapacitor={1/(2*PI*((550*FilterResistor)/(550+FilterResistor))*FilterCutoffFreq)}


ADC
.param fs = 1MEG              --> ADC Sampling rate. For ADAQ23875: min 20 ksps, max 15 Msps
.param ACQ_delay 100n         --> Delay from simulation start to first sampling request to the AD

PGA
.param PGIA_Gain=16  --> Instrumentation Op Amp gain. Valid values are 0 (shutdown), 0.25, 0.5
```

Figure 5.- TRAN example: other signal chain parameters

We are now ready to press the "Run" button to start the simulation.

Probing the input and output of the signal chain, we will get the desired plot. As we have a differential input, we will click on the positive terminal and drag towards the negative one to plot their voltage difference (green signal in the plot below).



Cursor 1
V(Vin_p,Vin_n)
Horz: 999.76232µs Vert: 249.99895mV
Cursor 2
Horz: -- N/A-- Vert: -- N/A--
Diff (Cursor2 - Cursor1)
Horz: -- N/A-- Vert: -- N/A--
Freq: -- N/A-- Slope: -- N/A--

Figure 6.- TRAN example: input and output signals

In Figure 6 we can confirm that the 500 mVpp signal has been amplified to 8 Vpp as expected.

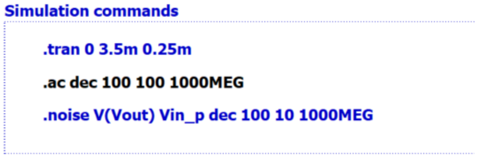
AC simulation

For the AC analysis, the simulation calculates the circuit response over the frequency domain.

AC simulation example: gain stage and signal filter bandwidth

In this example, we use an AC simulation to check the resulting bandwidths of the amplifier stage and the signal filter stage.

Commence by enabling the AC simulation command only:



Simulation commands

```
.tran 0 3.5m 0.25m
.ac dec 100 100 1000MEG
.noise V(Vout) Vin_p dec 100 10 1000MEG
```

Figure 7.- AC simulation enabled

We will configure the signal chain as in the previous example. As per its datasheet, the LT6373 has a 3 dB bandwidth between 4 MHz and 7.5 MHz depending on gain, so the 100 kHz signal filter is well within its passband.

After running the simulation, let's probe the ADC input (differentially, by clicking and dragging from positive to negative node) and the ADC output. The plot we get by default is the following:

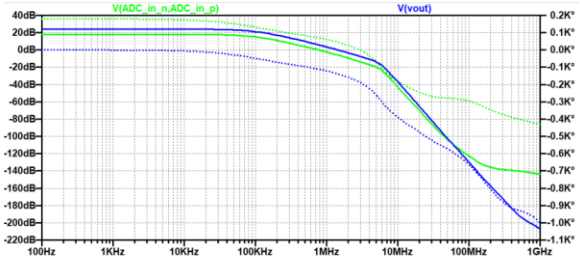
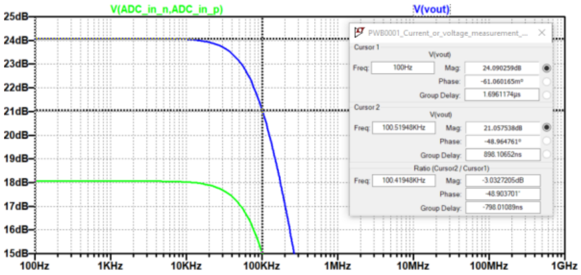


Figure 8.- AC example: ADC input and output default plot

Let's produce a more usable plot for our purpose by 1) right-clicking on the right y-axis, then clicking on the "Don't plot phase" button to remove the phase traces, 2) right-clicking on the left y-axis to adjust the range top, bottom and tick values (25dB, 15dB and 1dB, respectively) and 3) clicking twice on the V(vout) trace name to place two cursors and drag them until we get a measure of the 3 dB bandwidth of the signal chain. The result is this plot:



Cursor 1
V(vout)
Freq: 100Hz Mag: 24.090259dB ●
Phase: -61.060165m° ○
Group Delay: 1.6961174µs ○

Cursor 2
V(vout)
Freq: 100.51948KHz Mag: 21.057538dB ●
Phase: -48.964761° ○
Group Delay: 898.10652ns ○

Ratio (Cursor2 / Cursor1)
Freq: 100.41948KHz Mag: -3.0327205dB
Phase: -48.903701°
Group Delay: -798.01089ns

Figure 9.- AC example: ADC input and output formatted plot

Through the V(out) trace, that shows the transfer function of the whole signal chain from input to output, we confirm that we correctly set the 3 dB bandwidth to 100 kHz (as cursor 2 can be seen sitting at an amplitude 3 dB below that of the passband, and a frequency of 100 kHz). We can verify as well our intended gain of 16 (24.08 dB), which is the amplitude shown by the cursor sitting on the passband.

We can observe the amplitude at the input of the ADAQ23875 (green trace), to be 6 dB below of that of V_{out}, in the passband region. This is due to the interaction between the external RC filter and the FDA stage inside the ADAQ23875. For more details about this loading effect and how the signal filter bandwidth is calculated, see the section below.

Noise simulation

A noise simulation performed through the .NOISE command allows to extract the noise spectral density at a specified (output) node in the circuit. It is concerned with random noise (thermal, flicker, shot) generated by the components that comprise the circuit, meaning it has nothing to do with other kinds of unwanted signals such as coupled interference, out of band signal components, crosstalk, power supply harmonics, etc. The .NOISE analysis is a particular case of small-signal AC analysis, and it is independent from .TRAN and .AC ones. That is, even though .NOISE analysis exposes noise voltages present in the circuit, those voltages cannot be observed in the time domain through a .TRAN simulation or have any effect on an .AC simulation.

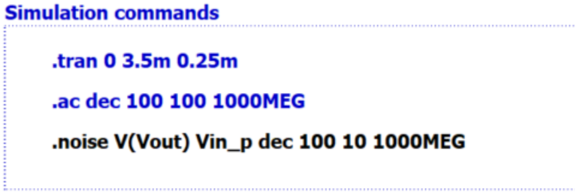
When performing full signal chain noise analysis, it is a good practice to check beforehand what is modeled in each individual component. In this case, the propagation of the noise present in the ADAQ23875 reference input to its output is not modeled.

For a first-order theoretical approach to signal chain noise analysis, see [7].

Noise simulation example: total signal chain noise

In this example, we will use the .NOISE simulation to check overall noise performance of the complete signal chain.

We start out by commenting out all simulation commands except for the .NOISE one:



Simulation commands

```
.tran 0 3.5m 0.25m
.ac dec 100 100 1000MEG
.noise V(Vout) Vin_p dec 100 10 1000MEG
```

Figure 10.- NOISE simulation enabled

The .NOISE simulation command requires to specify what is the circuit node where we want to measure the noise (in this case the output of the ADC) as well as the input signal source.

For this simulation run, we will configure the signal chain as in the previous AC simulation example.

The result we get in the waveform viewer after running the simulation, then left clicking on the output node is the following:

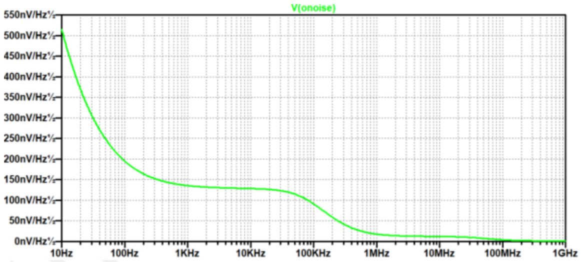
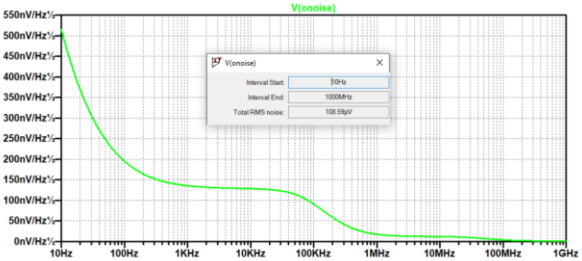


Figure 11.- .NOISE example: noise spectral density at signal chain output

This is the noise power spectral density at the output of our signal chain. The RMS value of the noise voltage is calculated by integrating the power spectral density over the bandwidth of interest. We can have the waveform viewer perform this integration for us by pressing CTLR+left click on the trace label:



V(onoise)
Interval Start: 10Hz
Interval End: 1000MHz
Total RMS noise: 108.59µV

Figure 12.- .NOISE example: displaying integrated RMS noise

As we can see, according to the simulation the digital signal at the output of this signal chain will exhibit 108.59 µVrms of noise at its output.

Note that the result is the integrated RMS noise of the **displayed** waveform. This means the result will change if we modify the frequency interval over which the analysis is performed (in the simulation command parameters), and it will also vary if we zoom-in on the x-axis.

Other considerations

Component models

LTspice models for all electronic components in this signal chain are included in the LTspice built-in library, so no external files are needed. Make sure your LTspice installation is up to date the by clicking on "Sync Release" from the "Tools" menu.

Simulation models for components don't necessarily cover the full behavior of the real device. In general, it is safe to assume that the component will display the correct behavior under standard operating conditions (room temperature, nominal supply voltage…) while second order effects such as distortion, crosstalk, etc. might not be included in the models.

For ADCs, the models focus on analog behavior so most of the component behavior, pins, etc. related to digital I/O are not present in the models. The LTspice model will not be giving out the coded output over a digital interface (SPI, LVDS) as the actual device does. Instead, the ADC output is provided as a regular (analog) signal that is quantized and restricted to the maximum input span of the ADC. This makes it easy to seamlessly perform signal chain analysis, especially noise and AC, up to and including the ADC output (which is a digital domain signal) in analog units of volts and hertz. For TRAN analysis, note that transient variations in the voltage reference of an ADC may not have a clearly identifiable effect in this analog version of the output, even though they could indeed cause code changes in the digitally coded output of the real device.

Simulation speed

Full signal chains might experience long simulation times. First because the signal chain will be composed by multiple components, but also because ADCs and DACs can have complex models, so it is not uncommon for them to be the main cause for a slow simulation. To alleviate this, consider removing the ADC temporarily from the schematic when exploring aspects of the signal chain that are not impacted by it.

For information on the topic of speeding up simulations see this article [8].

FDA gain and signal filter bandwidth

In our signal chain we have a programmable gain instrumentation amplifier (PGIA) driving a fully differential amplifier (FDA) through a differential RC filter, with the following topology:

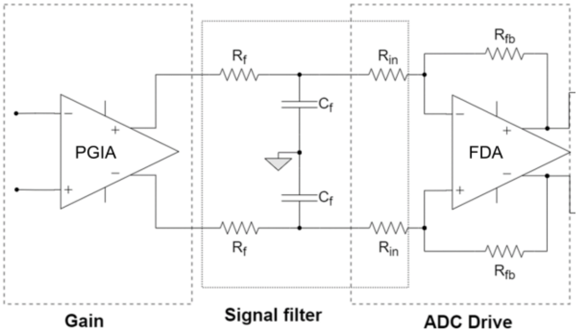


Figure 13.- PGIA driving an FDA through a differential RC filter

For determining DC/low frequency gain from PGIA output to FDA output, we can consider the C_f capacitors as open circuit and apply the regular expression for the gain of an FDA, which results in:

$$G = \frac{R_{fb}}{R_{in} + R_f}$$

For the gain of 1 that we've been wanting to achieve, this means:

$$R_f = R_{in} = \frac{R_{fb}}{2}$$

Justifying our selection of 550 Ω for the filter resistors.

Now, with regards to the bandwidth of the signal filter (and hence of this whole circuit), the double negative feedback in the FDA will force a virtual short circuit between its inverting and non-inverting inputs. So, the impedance seen looking into the ADC drive stage (as defined by the dotted line in Figure 13) is the series combination of both R_{in} resistors. This means that the R_{in} resistors are effectively in parallel with the C_f capacitors. This causes the response of the formed first order RC filter to be dictated by an effective resistance of $R_f||R_{in}$. The 3 dB cut-off frequency for this filter is therefore given by:

$$f_{3dB} = \frac{1}{2\pi \cdot (R_f||R_{fin}) \cdot C_f}$$

We can then solve the expression for C_f to be able to calculate the capacitor from a desired 3 dB cut-off frequency, as is done in the LTspice schematic with the *FilterCapacitor* parameter:

$$C_f = \frac{1}{2\pi \cdot (R_f||R_{fin}) \cdot f_{3dB}}$$

The loading effect from the R_{in} resistors is the cause for the observation we made in the AC simulation example, where the signal at the input of the ADAQ23875 appeared 6 dB attenuated dB (half amplitude) with respect to PGIA output. The following diagram illustrates this:

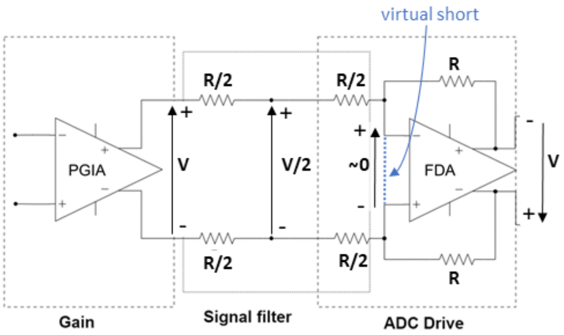


Figure 14.- Differential voltage divider and virtual short

Where the input of the ADAQ23875 is where the differential voltage of V/2 was measured, and the capacitors have been removed as we are focusing on DC/passband behavior, well below the 3 dB cutoff calculated above.

References

[1] G. Alonso, "Get Up and Running with LTspice," Analog Devices, [Online]. Available: https://www.analog.com/en/analog-dialogue/articles/get-up-and-running-with-ltspice.html.

[2] Analog Devices, "ADI EngineerZone LTspice forum," [Online]. Available: https://ez.analog.com/design-tools-and-calculators/ltspice/.

[3] Analog Devices, "Precision wide bandwidth signal chain: optimized noise, bandwidth and area," [Online]. Available: https://www.analog.com/en/applications/technology/precision-technology/precision-wide-bandwidth.html.

[4] P. Karantzalis, "Webinar: Signal Chain Filtering: Beginning with the Basics," [Online]. Available: https://www.analog.com/en/education/education-library/webcasts/signal-chain-filtering-beginning-basics.html.

[5] G. Alonso, "LTspice: Using the .STEP Command to Perform Repeated Analysis," [Online]. Available: https://www.analog.com/en/technical-articles/ltspice-using-the-step-command-to-perform-repeated-analysis.html.

[6] G. Alonso, "LTspice: Generating Triangular & Sawtooth Waveforms," [Online]. Available: https://www.analog.com/en/technical-articles/ltspice-generating-triangular-sawtooth-waveforms.html.

[7] R. Delaney and P. Delizia, "Step-by-Step Noise Analysis Guide for Your Signal Chain," [Online]. Available: https://www.analog.com/en/analog-dialogue/articles/step-by-step-noise-analysis-guide-for-your-signal-chain.html.

[8] G. Alonso, "LTspice: Speed Up Your Simulations," [Online]. Available: https://www.analog.com/en/technical-articles/ltspice-speed-up-your-simulations.html.