Signal chain LTspice simulation guide

Sinewave Voltage Generation Performance Optimized

Introduction

This guide offers a walkthrough on simulating a sinewave generation signal chain, optimized for performance, by 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 the ADI EngineerZone website [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.

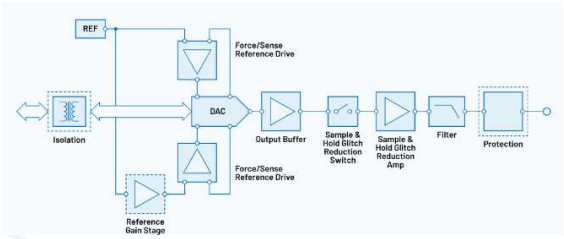

Figure 1.- Signal chain block diagram

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 20-bit, 1 ppm AD5791 Voltage Output DAC in combination with the very low noise LTC6655LN-5 voltage reference. A suite of high stability and low distortion opamps with very low noise were selected: the ADA4077 as the inverting amplifier for the negative reference, the AD8676 dual as the unity gain Kelvin-connected drivers for the positive and negative reference inputs to the AD5791 DAC, and the AD8676 dual opamp as the DAC output voltage buffer and deglitching amplifier. The AD8675 serves as the interpolation/reconstruction smoothing filter at the output. The output is further protected against external over-voltages with the ADG5421F protection switch.

The DAC itself produces no more noise than the 7.5 $nV/\sqrt{Hz}$ thermal noise of its 3.2 kΩ output resistance. This requires careful filtering of the reference at its NR pin, as well as the reference inverting amplifier feedback resistor to avoid additional noise.

In Figure 2 we can see a screenshot of the signal chain implemented in the LTspice schematic.

Signal chain LTspice simulation guide

Sinewave Voltage Generation

Performance Optimized

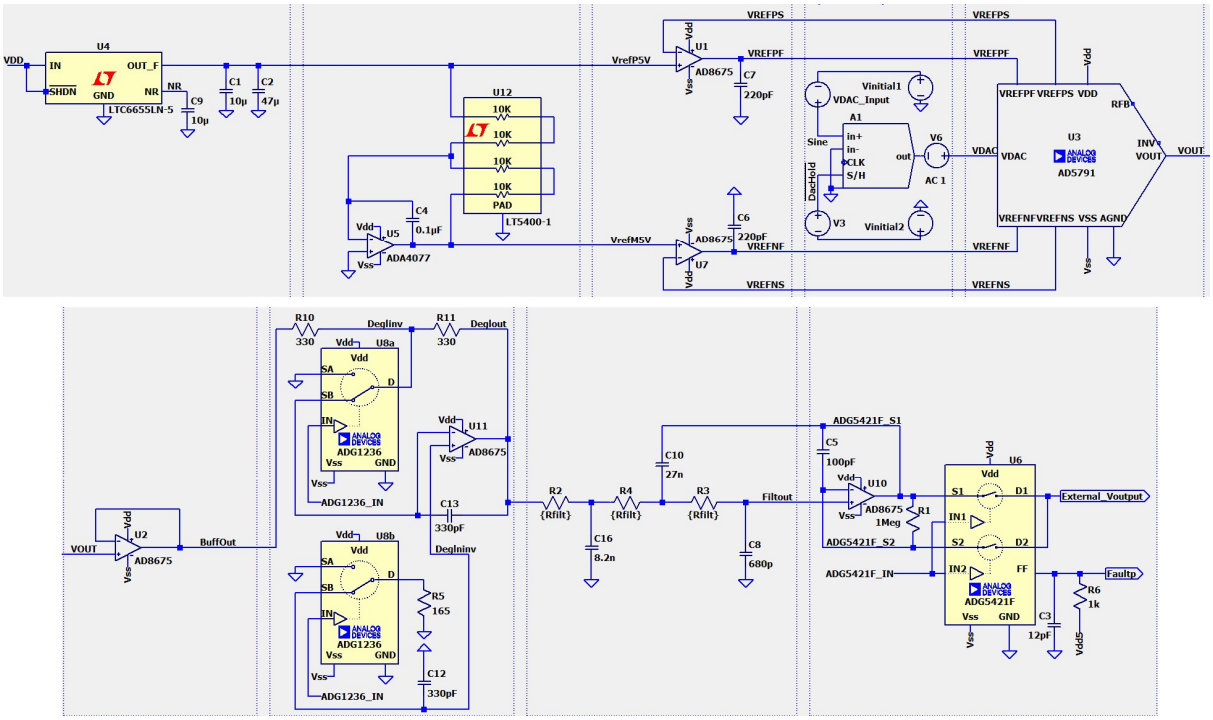

Figure 2.- Signal chain LTspice schematic

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 [4]). For example, the following parameters could be varied in a .STEP statement:

1. VDAC_DC=1 sets the DC operating point into the AD5791 DAC for AC and NOISE analysis. This parameter will be stepped later in this guide to show its effect on output noise.
2. Rfilt=2k sets the reconstruction filter bandwidth. The default 17 kHz bandwidth could be reduced to 8.5 kHz by increasing Rfilt to 4k.
3. Fs=100k sets the DAC update/sampling rate. A faster sampling rate creates a smoother sinewave, but at the potential expense of increasing distortion because each DAC transition is a non-linear event and may add harmonic distortion if not fully settled before the deglitching operation. Be sure that the minimum data sheet settling time of the DAC is maintained at the chosen update/ sampling rate. The maximum DAC Update Rate is (24 * 28 ns)+T6+T14 + settling time. The settling time depends on the time constant at DAC vout and on step size.
4. T=200/Fs is set to collect 200 samples of the input sinewave for subsequent FFT analysis in the waveform viewer.
5. Fin=2/T sets the input frequency to produce 2 complete cycles during the T=2ms duration of the saved data for subsequent FFT analysis in the waveform viewer. This parameter should be varied as integer cycles during T. For example, 3 kHz input frequency would be obtained with Fin=6/T.

Be sure to always have complete sinewave cycles saved during the .TRAN simulation. This guarantees that the signal at the input frequency and its harmonics will be represented by single lines in the FFT spectrum. If incomplete sine cycles are saved during .TRAN, the peaks will show wide skirts.

```
Parameters
  Supplies
.param vdd=15          ---> set high positive analog supply voltage
.param vss=-15         ---> set high negative analog supply voltage
.param vdd5=5          ---> set low positive analog supply voltage
.param vss5=-5         ---> set low negative analog supply voltage
.param vcc=5           ---> set logic supply voltage
  Signal Chain
.param VDAC_DC=1       ---> set DAC equivalent DC input voltage. Use
.param Rfilt=2k        ---> set resistor value for reconstruction/inte
.param Fs=100k         ---> set DAC update (sampling) frequency Fs i
.param T=200/Fs        ---> set T time record from 200 samples of 1/
.param Fin=2/T          --> set input frequency for 2 cycles during T.
  Output Protection
.param ADG_IN=vcc      ---> enable ADG5421F protection switch
```

Figure 3.- Signal chain parameters

Control signals

The operating mode of the output overvoltage fault protection switch ADG5401F is controlled by pin IN. The state of this signal is set in the .PARAM block as ADG_IN=Vcc to enable the switches and connect the signal chain to the outside.

The AD5791 DAC model

VDAC is an equivalent voltage representing the digital DAC input:

Vdac=0V corresponds to 0x00000 (zero scale) Vout=VrefP5V=-5V

Vdac=0.5V corresponds to 0x80000 (zero scale) Vout= VrefM5V /2+ VrefP5V/2=0V

Vdac=1V corresponds to 0xFFFFF (full scale)

Vout=VREFPS=+5V

The DAC output VOUT responds to the reference voltages and the input VDAC with this transfer function:

$$VOUT = VrefP5V \times \frac{VDAC}{2} + VrefM5V \times \frac{(1 - VDAC)}{2}$$

The DAC model is a purely linear continuous time model, without quantization or sampling and will respond continuously to voltages inside and outside the 0V-1V range. The A1 sampling block emulates a zero order hold function that samples its input into a series of steps that are updated at the Fs=100 kHz default) rate.

Note that the VDAC input into the DAC merely controls the proportion of the positive and negative references that reaches VOUT. This is the fundamental behaviour for the architecture of

the AD5791 DAC. The positive reference at VrefP5V is +5V and the negative reference voltage at VrefM5V is -5V. The mid-scale output voltage with VDAC=0.5, which corresponds to a digital code of 0x80000, is the average of the positive and negative reference voltages. Any mismatch between the positive and negative reference voltages causes an offset voltage at the DAC output Vout. The precision resistor network U12 in the feedback path of U5 has a maximum mismatch error of 0.01%, which contributes less than ±125 µV offset error at the DAC output VOUT. If digital system calibration is used, a cancelling offset could be added to the data input of the DAC to eliminate any remaining offset error.

Transient simulation

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

In the following examples we will use transient simulations to check powerup settling, step settling, sinewave burst settling and the collection of complete sinewave cycles for subsequent FFT analysis in the waveform viewer.

Three .TRAN simulation commands can be found in the *Simulation commands* section of the schematic, as can be seen in Figure 4. The .TRAN simulations are set to last 200msec to let the LTC6655LN-5 reference and the inverting reference amplifier U5 stabilize at VrefP5V and VrefM5V respectively, after the initial power-up. Each version will help focus on different aspects of the signal chain operation.

```
.tran 0 200m 0 startup
.tran 0 {200m-T} {200m-T-2m} startup
.tran 0 200m {200m-T} startup

.ac dec 100 1 1e9

.noise V(External_Voutput) VDAC_Input dec 100 1 1e9
```

Figure 4.- Simulation commands section

Transient simulation example: Powerup settling

For this example, the first TRAN command is enabled while all the rest are commented out (like Figure 4 depicts). This command simulates the entire 200 ms period and includes:

1. 0s-196.5ms : Settling of reference voltages
2. 196.5ms-197ms : 5V step settling
3. 197ms-200ms : Three cycles of 1 kHz sinewave burst

This full-length 200 ms simulation is useful to try probing several signals in the schematic and to zoom into different portions of the signals to familiarize yourself with the circuit operation.

After running the simulation and probing the VrefP5V and VrefM5V nodes, we get the result in Figure 5. It shows reference settling of the VrefP5V and VrefM5V nodes after powerup. You can zoom-in or assign the cursors by right-clicking the name of each signal, then left-click the cursor lines to move them around.

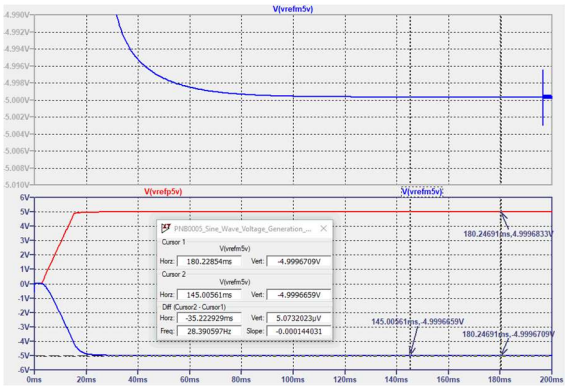

Figure 5.- TRAN example: VrefP5V, VrefM5V powerup settling

Read cursor values and their differences in the cursor box, or label cursor locations with the F4 key or through the menu: PlotSettings→ Notes&Annotations→LabelCursPosition.

The cursor values and labeled locations on the second plot of Figure 5 show that VrefM5V is within 5 µV of its final value at 145 ms. There is

less than 1 µV settling error after the 180 ms cursor location. The first plot zooms into the VrefM5V trace to show detail in the settling behaviour. VrefM5V is driven by VrefP5V and thus settles last. We will explore next the last 4 ms of 200 ms simulation with the second and third .TRAN commands.

Transient simulation example: Settling of 5V step, Deglitching, Filtering

Now comment out the first .TRAN command and turn the second one into a "SPICE directive" as was done for the previous simulation.

The second .TRAN command in Figure 4 simulates for 198 ms and saves the last 2 ms, starting at 196 ms to focus on the -5 V to 0 V step settling portion and the first cycle of sinewave settling after the 0 V input. Figure 6 shows the response of a 5 V step and single 1 kHz sine cycle at the DAC output V(Vout), the deglitcher stage output V(Deglout) and the filtered signal chain output in the V(External_Voutput) trace.

The Sallen-Key 3rd order reconstruction filter is the slowest settling stage in the chain as seen in the second panel in the step response of V(External_Voutput). The 5 V step settling to 5 µV takes 842 µs - 500 µs = 342 µs, at the labeled point. One can expect that settling into the sine cycle takes less than that time. Note the delay of the filtered V(External_Voutput) with respect to the unfiltered input at V(Deglout).

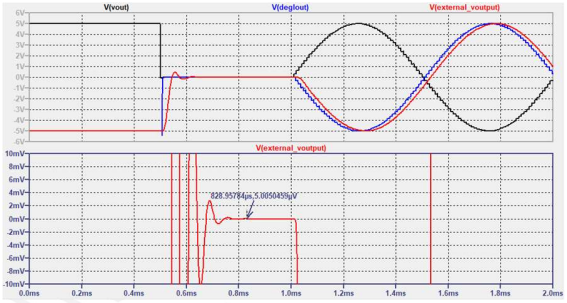

Figure 6.-TRAN example: 5V step and sinewave settling

Figure 7 shows zoomed-in detail of the unfiltered V(Vout) in black, the deglitched and inverted -V(Deglout) in blue and the filtered output -V(External_Voutput) in red. These two traces are plotted as negatives to appreciate the relationship with V(Vout). The staircase pattern illustrates the classic zero-order hold signal from the DAC. The pink/magenta V(ADG1236_IN) is the control signal for the deglitcher circuit. When it is high, the output of the deglitcher goes into hold mode, so that the DAC can update to the next level. The narrow pulses in the green V(DacHold)-2 clock trace update the DAC input VDAC just after V(ADG1236_IN) goes high which, in turn, produces the black V(Vout) trace. After the DAC is settled in the black V(Vout) trace, V(ADG1236_IN) goes back low and the deglitcher circuit in the blue -V(Deglout) trace updates **linearly** to the latest V(Vout) value. The red trace is the filtered output -V(External_Voutput) and thus settles much more slowly.

The benefit of the deglitcher circuit is to blank out the non-linear behaviour that occurs during the transition of the DAC output to a new value. The glitch area during DAC code transitions is a nonlinear function of the DAC input. This distortion cannot be filtered by the 17 kHz bandwidth reconstruction filter and would compromise the Total Harmonic Distortion (THD) of the sinewave. Keep in mind that the DAC model is purely linear and does not exhibit code transition distortion. You will notice small negative-going glitches in the blue -V(Deglout) trace, but this glitch is constant and does not carry any distortion products.

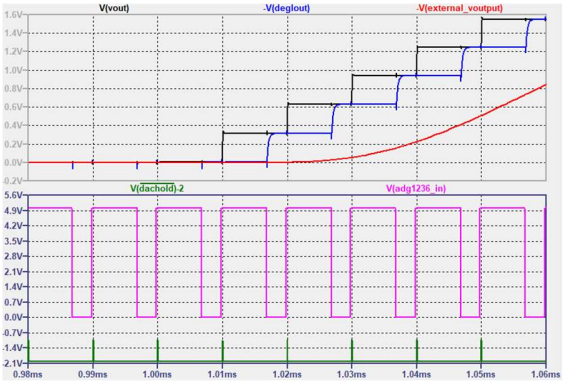

Figure 7.-TRAN example: DAC staircase, deglitching, filtering

Transient simulation example: Sinewave collection for FFT analysis

To run this example, the only active simulation directive must be the third and last .TRAN command.

The third .TRAN command simulates for 198 ms to let the reference voltages settle out from power-up and saves the last 2 ms of a total 3 ms 1 kHz sinewave burst up to 200 ms. See Figure 8. These 2 ms contain exactly two cycles of a 1 kHz sinewave updated by the DAC at a 100 kHz update rate with the default schematic parameters. The 100 kHz update rate is also known as 100 ksps (kilo samples per second). Because this 2 ms simulation record contains a synchronous, settled and complete number of input sinewave cycles, you can view the FFT of any time domain trace that you plot.

The plot in Figure 8 shows the 0-1 V V(Sine) ideal analog sinewave in the black trace, the DAC output V(Vout) in the blue trace, the deglitched and inverted output V(Deglout) in the green trace and the filtered signal chain output at V(External_Voutput).

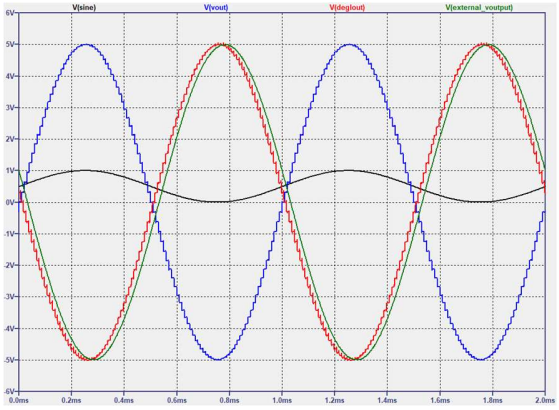

Figure 8.- TRAN example: 1 kHz input sinewave, 100 ksps

Once you plotted the signal in the viewer window and that window is active as your current window, select from the menu: View→FFT, click OK with the selected signals, then click the frequency domain signals in the next dialog for plotting and click OK again. Figure 9 shows the spectrum of 4 sinewave signals that were preselected from the time-domain plot. Each of the signals has an associated scale factor to bring the 1 kHz fundamental to 0 dB to compare the harmonic levels more easily. Keep in mind that the frequency resolution of the FFT is only 1/T=500 Hz. That is, the FFT only has data points every 500 Hz. The apparent skirts around the fundamental are just straight-line interpolations between frequencies, which get curved by the log-log scales of the plot.

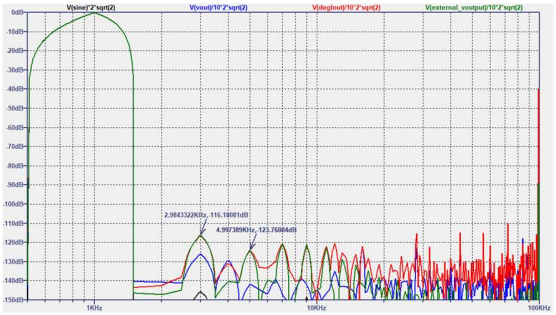

Figure 9.- FFT example: 1 kHz input sinewave, 100 ksps

V(Sine) is the ideal continuous analog 0-1V, 1 kHz sinewave input and is plotted here to verify that we have a complete number of sine cycles as the input to the FFT. If there was a partial cycle, large skirts would arise on either side of the 1 kHz fundamental and hide the low level of the harmonics. Then we have the spectra of the raw DAC output V(Vout), the inverted deglitched V(Deglout) and the filtered signal chain output V(External_Voutput). Notice the high harmonics in the red trace V(Deglout), which get filtered by the 17 kHz 3-pole filter and are not present in the green output trace V(External_Voutput).

The 3rd and 5th harmonic of V(External_Voutput) are probed with the cursor and marked with F4. The -116 dB and -123 dB harmonic distortion values for the 3rd and 5th harmonics are only illustrative because the DAC model for the AD5791 is exclusively linear. Other limitations to FFT accuracy include the time resolution of sharp transients, which were not saved with enough time resolution. You might try setting the "Maximum Timestep" in the simulation dialog to something like 1 ns, but that would dramatically slow down the simulation.

AC simulation

The AC analysis simulation calculates the circuit response over the frequency domain.

AC simulation example: DAC, Deglitcher and Filter Bandwidths

In this example, we will use an AC simulation to check the bandwidths of the DAC output, the deglitcher stage and the output reconstruction smoothing filter. The deglitcher is in track mode during AC analysis.

Commence by enabling only the AC simulation command as shown in Figure 10:

```
.tran 0 200m 0 startup
.tran 0 {200m-T} {200m-T-2m} startup
.tran 0 200m {200m-T} startup

.ac dec 100 1 1e9

.noise V(External_Voutput) VDAC_Input dec 100 1 1e9
```

Figure 10.- AC simulation enabled

The simulation is run with Rfilt=2k for a 17 kHz bandwidth at the 3rd order Sallen-Key reconstruction/smoothing filter.

After running the simulation and probing the Vout Deglout and External_Voutput nodes we get the result in Figure 11:

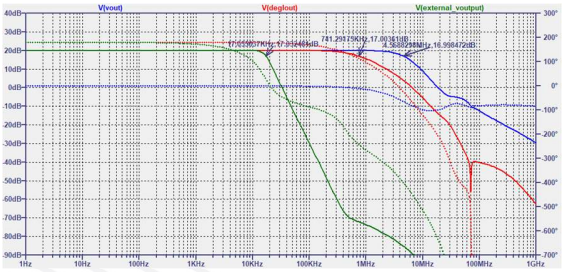

Figure 11.- AC example: measuring

We can verify the 3 dB bandwidth at the raw DAC output V(VOUT), the deglitcher stage at V(Deglout) and the overall signal chain filtered output at V(External_Voutput). Figure 11 marks these 3 dB points as 4.6 MHz, 741 kHz and 17 kHz respectively. The 20 dB apparent in-band gain results from the DAC model input which is 1 V_{pp} and appears as a 10 V_{pp} swing at the DAC ouput. The Deglitcher has a gain of -1 and the Sallen-Key filter has a passband gain of 1.

If you wish to see the signal chain bandwidth from starting points other than the VDAC input, you need to move the V6 AC source, which has an AC amplitude of 1 V, to the desired location. For example, you could open the VrefP5V connection between the +5V reference U4 and the U12 resistor network to see what the AC-response is, starting at the reference U4, all the way to the output. Keep in mind that the amount of VrefP5V vs VrefM5V that appears at the output is a direct function of the VDAC at the DAC input.

The transfer function from VDAC, VrefP5V, VrefM5V to VOUT is:

$$VOUT = VrefP5V \times \frac{VDAC}{2} + VrefM5V \times \frac{(1 - VDAC)}{2}$$

For example, with VDAC=0.5, VOUT is the average between VrefP5V and VrefM5V, which adds up to 0 V.

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.

These noise simulation examples apply for a steady DAC input and are thus also free from DAC quantization noise or any kind of non-linearity induced noise.

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 from 0.1 Hz to 1 GHz.

We will start by commenting out all simulation commands except for .NOISE, see Figure 12:

```
.tran 0 200m 0 startup
.tran 0 {200m-T} {200m-T-2m} startup
.tran 0 200m {200m-T} startup

.ac dec 100 1 1e9

.noise V(External_Voutput) VDAC_Input dec 100 0.1 1e9
```

Figure 12.- NOISE simulation enabled

Note that the simulation command requires specification of the circuit node where we want to measure the noise, for example at External_Voutput, as well as an input signal source, for example at V6. Keep in mind that the noise that may come in through the VrefP5V and VrefM5V DAC inputs is a function of the input code at VDAC.

This means that at midscale (VDAC=0.5, VOUT=0V) the noise coming from the LTC6655LN-5 is cancelled to the extent that it appears equally with opposite phase at VrefP5V and VrefM5V. The low frequency noise of the LTC6655LN-5 is 0.6 μV_{pp} between 0.1 Hz and 10 Hz and is included in the model. The 75 $nV/\sqrt{Hz}$ thermal broadband noise of the LTC6655LN-5 is also in the model.

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

Figure 13 shows the output noise at V(onoise) in the waveform viewer after running the simulation.

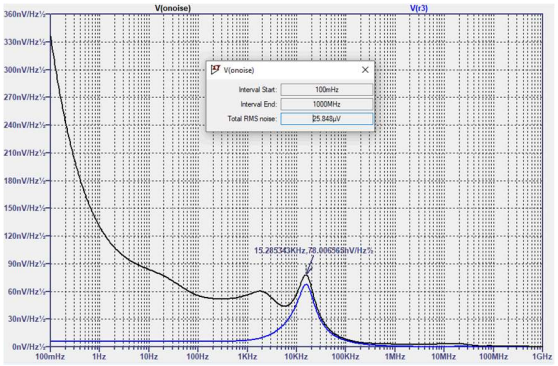

Figure 13.- NOISE example: for VOUT=+5V

This is the noise power spectral density at the output of our signal chain. The waveform viewer can perform for us its integration by pressing CTLR+left click on the trace label V(onoise). The integrated rms noise in the inset box is 25.6 µV. You may also right-click the Y-axis to change to log scale to view the various filter frequency corners more clearly.

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

The Sallen-Key filter has a noise peak caused by R3=2k at 15.5 kHz. The total noise from R3 is plotted in the blue trace of Figure 13. If we integrate the rms noise for R3 with CTLR+left click on V(R3) we get 9.6 µV.

Now we zoom around the noise peak from about 600 Hz to 60 kHz. The integrated noise contained in the peak is 10 μV_{rms}. This is commensurate with the total wideband noise from R3 from Figure 13. We can calculate a subtraction of the noise contribution from R3 as:

$$\sqrt{25\ \mu V^2 - 10\ \mu V^2} = 23\ \mu V$$

So, the noise peak only adds 2 µV to the total noise. The noise peak looks worse than it is,

because of the frequency log scale, which exaggerates the width and area in the peak.

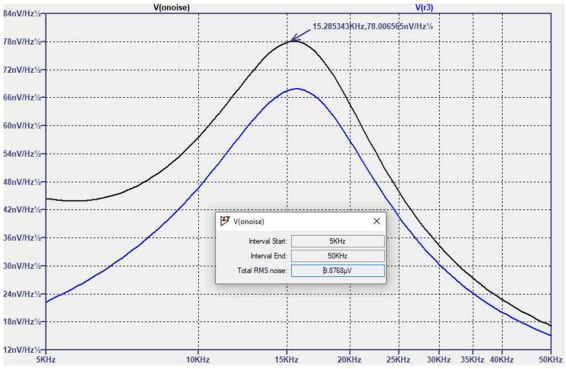

Figure 14.- NOISE example: for VOUT=+5V, Noise peak

As pointed out in the DAC transfer function, the content of each reference input is proportional to the input code and consequently to the output voltage. We use the .STEP command over the VDAC_DC parameter to iterate the noise simulation for VDAC inputs equal to 0 V, 0.25 V, 0.5 V, 0.75 V and 1 V.

```
.noise V(External_Voutput) VDAC_Input dec 100 0.1 1e9
.step param VDAC_DC list 0 0.25 0.5 0.75 1
```

Figure 15.- NOISE example: .STEP command

The corresponding DAC output voltages are -5 V, -2.5 V, 0 V, +2.5 V and +5 V. Right-click on the .STEP command and click the “SPICE directive” to enable this command and run the simulation.

The stepped curve family for V(onoise) as a function of the DAC input parameter VDAC_DC is shown in Figure 16. The @1, @2, etc suffixes after V(onoise) represent the number of the step. You may simply probe V(onoise) without a suffix, and you will get the full step family. To determine which curve goes with which step, you can right-click on the plot window area and select *View→ Step legend* to bring up a legend window with this information. Stepped curves cannot be integrated to calculate total noise; in order to get those results you must simulate these curves individually by commenting the .STEP command as shown above for other commands and manually assign values to the VDAC_DC parameter in the .PARAM statement.

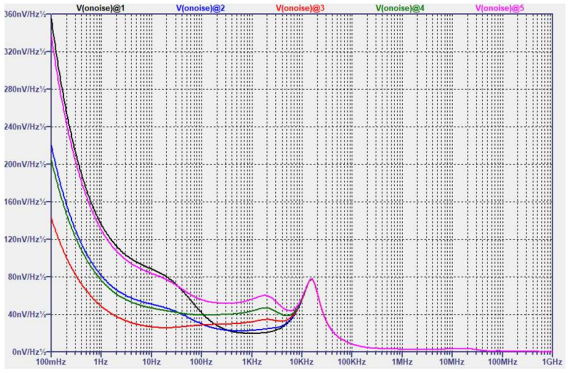

Figure 16.- NOISE example: stepped run for VOUT=-5 V, -2.5 V, 0 V, +2.5 V and +5 V

The reference noise below 100 Hz is strongly proportional to the VOUT voltage. However, this dependency contributes less than 0.5 μV_{rms} to the total noise because it is heavily band-limited by the reference filter capacitors.

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 do not 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.

As LTspice is not a mixed signal simulation environment, for DACs most of the component behavior, pins, etc. related to digital control and output are not present in the models. You will find that the “digital” input is a regular (analog) signal that is scaled to represent the digital input to the DAC, that is, the LTspice model does not accept binary codes over a digital interface (SPI, LVDS).

.TRAN Simulation speed and accuracy

The simulation accuracy for the subsequent FFT calculation in the viewer is limited by the accuracy of the very sharp transients that occur when the DAC is updated. This accuracy could be improved by reducing the “Maximum Timestep” parameter in the dialog box for the .TRAN simulation to something like 1 ns, but this dramatically increases simulation time in the 200 ms simulations.

Figure 17 shows a side stimulus circuit that produces underdamped 100 MHz transients just before each critical analog timing edge in the main circuit. This serves to slow down the timestep during transient simulation in anticipation of critical analog steps in the main circuit, to improve accuracy. This technique is a much faster alternative to constantly forcing a maximum timestep of 1 ns, because it remains off for the 196 ms power-up settling period and only slows down the simulation at the most critical moments.

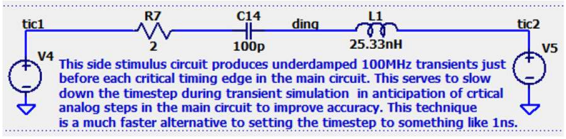

Figure 17.- Side stimulus improves sharp transient accuracy

For more information on the topic of speeding up simulations see this article [5]

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 narrow bandwidth signal chains: Sine Wave Voltage Generation," [Online]. Available: https://www.analog.com/en/applications/technology/precision-technology/precision-narrow-bandwidth.html#sine-wave.

[4] 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.

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

[6] Analog Devices, "Precision narrow bandwidth signal chains," [Online]. Available: https://www.analog.com/en/applications/technology/precision-technology/precision-narrow-bandwidth.html.

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