DAT116 - Omnibus

Simon Jansson, Andreas Johansson

5 January 2024

1 Lab 1 - Effects of sampling

In the first lab, Simulink was used to analyze the sampling of sinusoidal signals. Signals of multiple frequencies were combined to see how the frequency components would appear in the frequency spectrum when aliasing occurs. Gaussian noise and Butterworth filters were also used to see how to get the signal-to-noise ratio (SNR) and to improve it.

1.1 Uniform sampling

Firstly, a simple sine wave seen in Eq (1) had its time and frequency characteristics measured to understand the basics of what happens when signals are sampled in Matlab. The signal had an amplitude of 0.1, a frequency of 7 Hz, and was simulated over 1 s with fixed steps of $1/512s$ for a sampling frequency of $f = 512$ Hz. Fig 1 shows the setup in Simulink where the box to the right of the figure shows the signal name in the workspace where a vector containing the sampled signal values will be analyzed further in later parts of the lab.

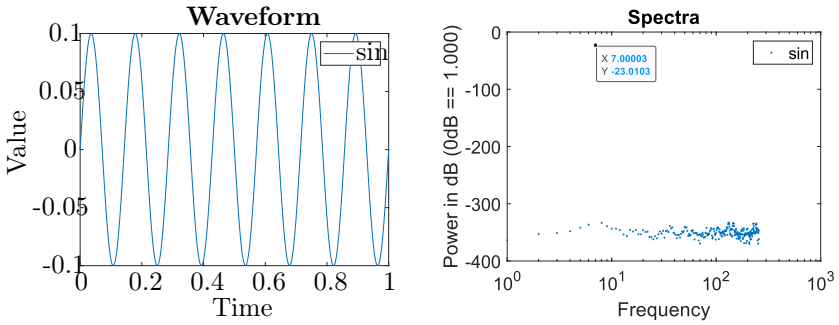
$$y = 0.1sin(2\pi * 7t) \tag{1}$$

Figure 1: Simulink model of sinus wave

Fig 2 shows the time and frequency characteristics of the signal. In the frequency spectrum, Fig 2b, the x-axis is the frequency, and the y-axis is the power. It shows one deviating data point with a frequency of 7 Hz and a power of -23 dB which is considerably larger than the other data points. This deviating value is the sine wave input previously described since it was defined to be a 7 Hz signal. Its power corresponds to the analytical calculations done in preparation for the lab which can be seen in Eq (2).

$$\begin{aligned} P_{avg} = (y^2)_{avg} = \frac{1}{1-0}\int_0^1 0.1^2 sin^2(14\pi t)dt = 0.01\int_0^1 \frac{1-cos28\pi t}{2}dt = \\ 0.005[t - \frac{sin28\pi t}{28\pi}]_0^1 = 0.05(1-0-(0-0) = 0.05 = -23.01dB \end{aligned} \tag{2}$$

The frequency spectrum shows many signals with differing frequencies from the given input which are all very weak at a power of around -350 dB compared to unit power. This comes from it not being possible to use infinite decimal values to describe every sampled data point. Therefore the sampled value will be a little bit off the real value for the 7 Hz sine wave. The difference will be interpreted as very low power (since the difference is so small) signals added on top of the larger 7 Hz sine wave. Very small imperfections from arithmetic on floats also contribute.



(a) Time diagram (b) Frequency spectrum

Figure 2: $7Hz$ sine wave

1.2 Multiple sin waves

From the model of the previous task shown in Fig 1, three more sine waves of the same amplitude were added to make a new model shown in Fig 3. The new sin signals have a frequency of 18, 46, and 65 Hz respectively. The now four sine waves were then superimposed upon each other with a sum block. A scope was also added to be able to inspect the combined signal.

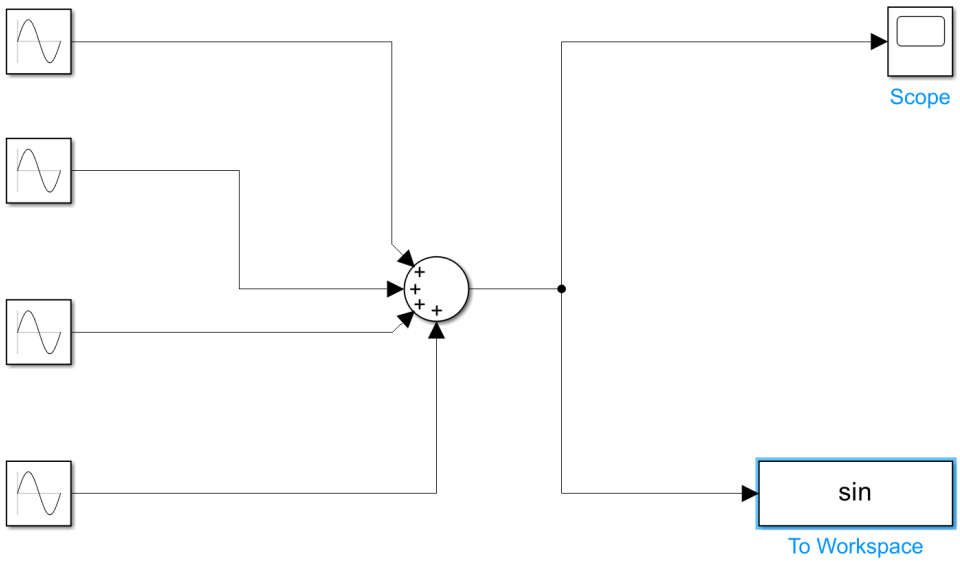


Figure 3: Simulink model of superimposed sine waves

The time diagram and frequency spectrum of the new model can be seen in Fig 4 and 5a respectively. They are both as expected since it is reasonable for the time domain diagram to look chaotic when there are multiple overlapping signals and the frequency spectrum has four distinct high power frequencies corresponding to the frequencies of the input signals.

Step by step the sample interval was increased by a factor of two-three times to produce in total four different frequency spectrums with f_s of 512, 256, 128 and 64 Hz respectively which are shown in Fig 5. With each halving of f_s the highest frequency shown in the frequency spectrum gets shrunk by half. This is because the spectrum shows the first Nyquist zone, or baseband, which goes from 0 Hz to the Nyquist frequency f_n defined as $f_n = f_s/2$. Fig 5a looks like Fig **??**, but with the extra high power frequencies signifying the new interpolated signals. Fig 5b is like Fig 5a, but with half the baseband. The time diagram of the original $f_s = 512Hz$ signal is shown in Fig 4.

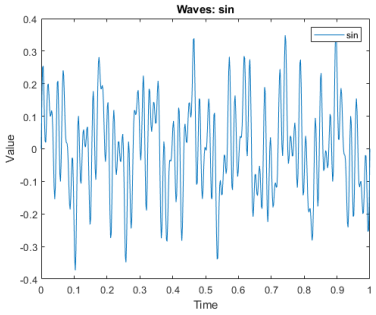
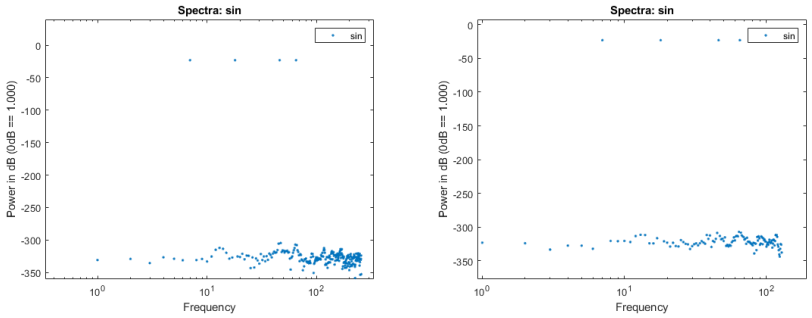
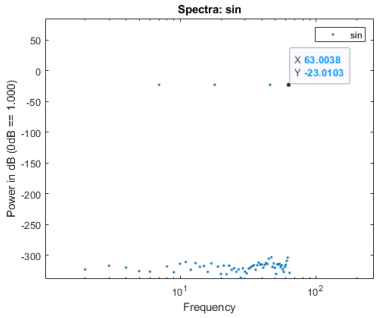


Figure 4: Time diagram when $f_s = 512Hz$ for combined signal

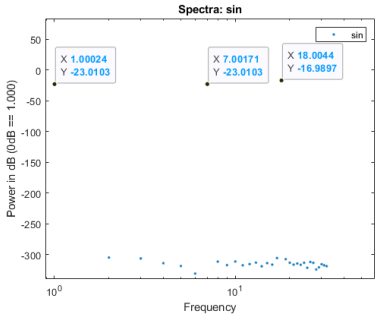


(a) $f_s = 512Hz$

(b) $f_s = 256Hz$



(c) $f_s = 128Hz$



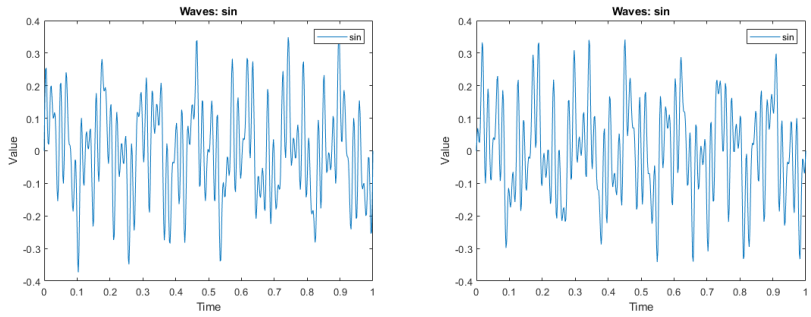
(d) $f_s = 64Hz$

Figure 5: Frequency spectrum of combined signal for different f_s

In Fig 5c the highest frequency spike is at 63 Hz. This might be considered strange since none of the input frequencies in the model have that frequency. The reason it appears is that Fig 5c has $f_s = 128Hz$ which means the Nyquist frequency is $f_n = f_s/2 = 128/2Hz = 64Hz$. Since the highest input frequency is 65 Hz, it is outside the baseband. Signals outside the baseband get aliased. I.e. the signal gets moved into the baseband f_n and since it is in an even Nyquist zone, zone two, it also gets mirrored around f_s. 65 Hz is 1 Hz above f_n which when aliased makes it 1 Hz below f_n. 1 Hz below f_n is 63 Hz which explains the mystery signal.

For the case of the slowest f_s, $f_s = 64Hz$, shown in Fig 5d, a lot of aliasing happens since it makes the two input frequencies 65 Hz and 46 Hz be above f_n. Since 65 Hz is more than f_s, the signal is in the third Nyquist zone. Signals there are directly offset into the Nyquist band which puts it at 1 Hz, since it is 1 Hz into the third Nyquist zone. The other frequency above f_n, 46 Hz, gets as what happened for the 65 Hz signal in Fig 5c when $f_s = 128Hz$, aliased into the baseband. This time though, there is already a signal with that frequency there, 18 Hz, whose power gets increased. Fig 5d shows the power at 18 Hz to be approximately -17 dB which is quadruple the power of -23 dB.

Lastly, the signal spectrum should be compared before and after the phase of the 46 Hz



(a) No phase shift (b) π phase shift

Figure 6: Interpolated signal in t with and without phase shift on $46Hz$ signal

signal has been increased by π. This has been done in Fig 6. As can be seen, the time diagrams are quite different with more large maxima early and more large minima later on after the phase shift. Since the phase has no effect on frequency, any change in phase will not affect the frequency spectrum unless it cancels out another signal. The frequency spectrum therefore shows no difference which is why it is not shown here. As has here been shown, the time domain can be changed quite drastically without a change in frequency by phase shifting.

1.3 Anti-aliasing

Continuing from the previous task an analog Butterworth filter was introduced to the system's output. The filter employed low-pass characteristics to shape the frequency response of the signal. The resulting system configuration is illustrated in Fig 3 below.

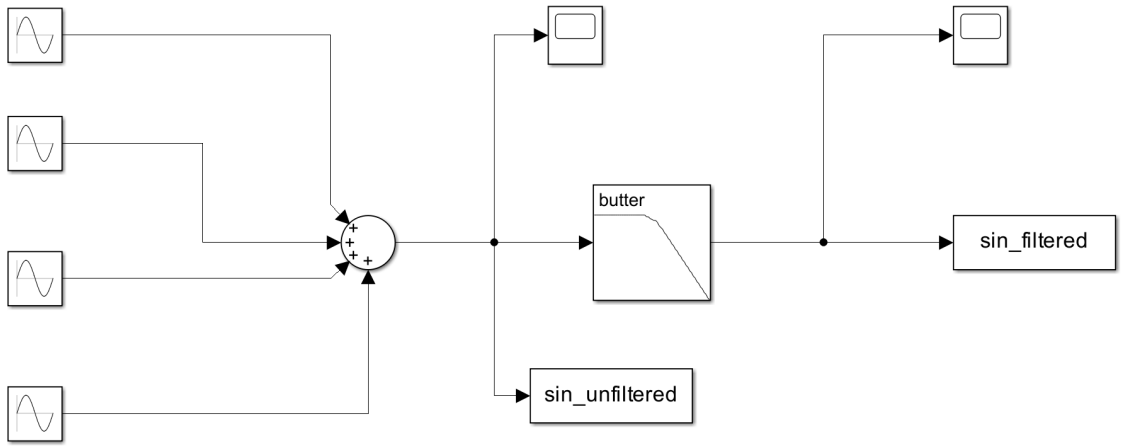
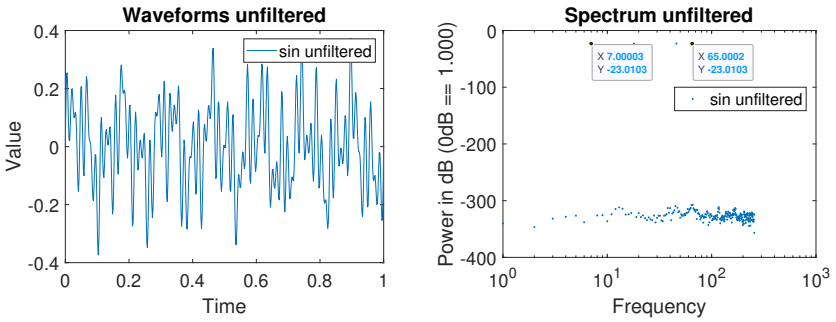


Figure 7: Simulink model with a filter

A Butterworth filter of the third degree is described by Eq (5) where $j\omega$ is the input frequency and ω_c is the angular cutoff frequency ($\omega = 2\pi f$).

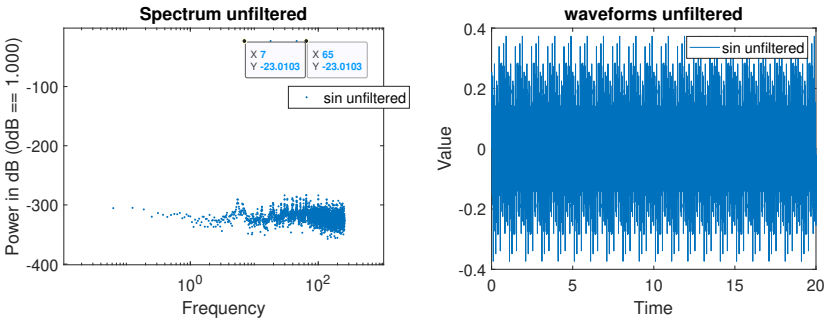
$$H(j\omega) = \frac{1}{\left(j + \left(\frac{\omega}{\omega_c}\right)\right)^{2n}} \tag{3}$$

When analyzing the filter, two distinct scenarios were examined: one involved a simulation duration of 1.9 seconds with 512 sample points, and the other spanned 20 seconds with 8192 points. The results for these scenarios are depicted in Figure 10 and Figure 11 respectively.



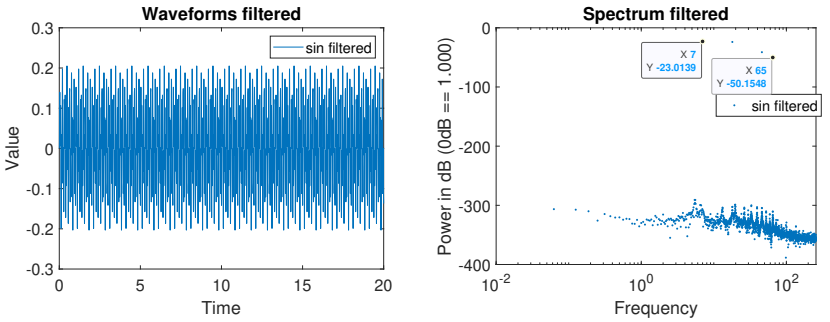
(a) Time diagram (b) Frequency Spectrum

Figure 8: Unfiltered signal for 1.9s simulation time



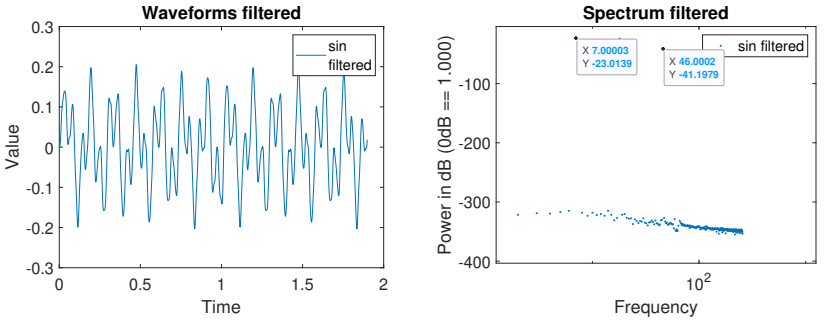
(a) Time diagram (b) Frequency spectrum

Figure 9: Unfiltered signal for 20s simulation time



(a) Time diagram (b) Frequency spectrum

Figure 10: Filtered signal for 20s simulation time



(a) Time diagram (b) Frequency spectrum

Figure 11: Filtered signal for 1.9s simulation time

From both scenarios, it can be inferred that the filter is at least operating. However, to establish its attenuation to be reasonable, the attenuation of the filter was manually calculated with Eq (4).

$$H(s) = 10 \cdot log \frac{P_{46}}{\sqrt{1 + \frac{f_s}{f_c}^6}} \tag{4}$$

For the $46Hz$ the attenuation of the filter was calculated in Eq (5) to be 32.1dB. This is quite similar to the attenuation recorded in the simulation which was 28 dB. Since some

loss in effectiveness is to be expected when simulating, the attenuation is as was expected.

$$10 \cdot \log\left(\frac{0.005}{\sqrt{1+\left(\frac{46}{23}\right)^6}}\right) = 32.1dB \tag{5}$$

1.4 Signal-to-noise ratio (SNR)

To investigate the effect of random noise a model was created with additive Gaussian noise being applied to the sine wave of Eq (1). The variance of the noise, which is its power, was set to be equal to the power of the sine wave. The model can be observed in Fig 12 and the output seen in Fig 13 for a simulation time of 1.9s. In Fig **??** the notated data point represents the power of the sine wave which is, as expected, clearly more powerful than the noise.

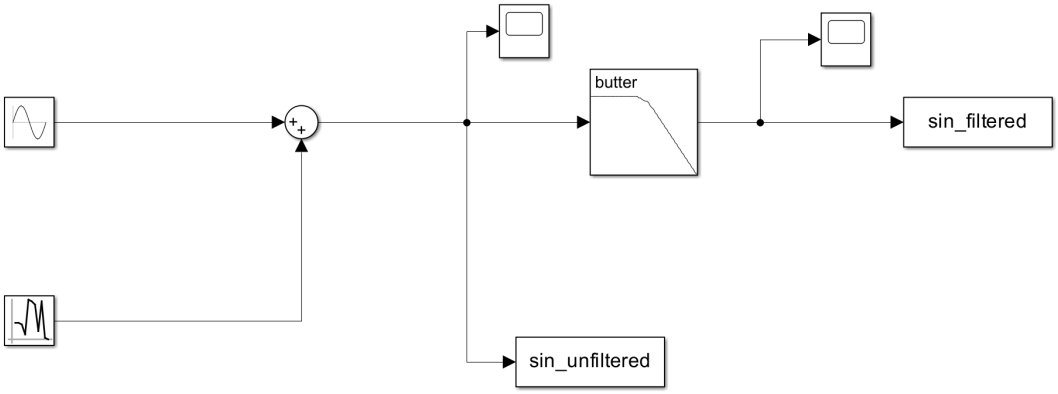
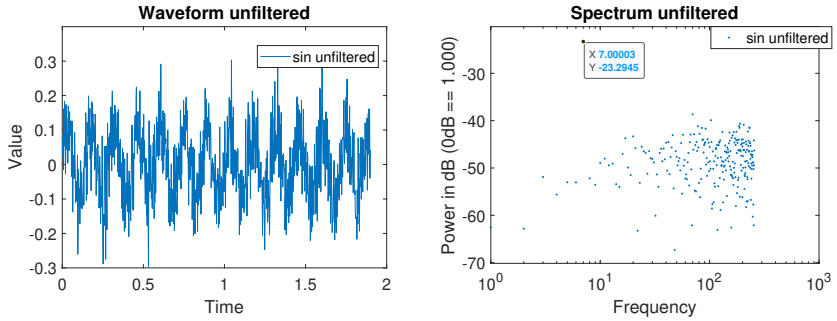


Figure 12: Simulink model of a sinus with a Gaussian noise

From the sampled filtered signal a vector containing the power of the frequencies is obtained. From this, the power of the desired signal is divided by the power of all of the signals attributed to noise as seen in Eq (6) to get the signal-to-noise ratio (SNR). In Eq 6 P_s is the power of the sine wave and P_i is the power of frequency i.

$$SNR = \frac{P_s}{(\sum_{n=1}^{\infty} P_i) - P_s} \tag{6}$$

This resulted in SNR =-0.58 dB which is as poor as expected since the power of the noise was set to be equal to the sine wave input. The reason why the noise is shown as more powerful by the SNR being negative is because of noise being applied to the sine wave frequency of $7Hz$. Fig 13b shows the power at $7Hz$ to be a little bit less at $-23.29dB$ rather than at $-23.01dB$ as it is in previous tasks.



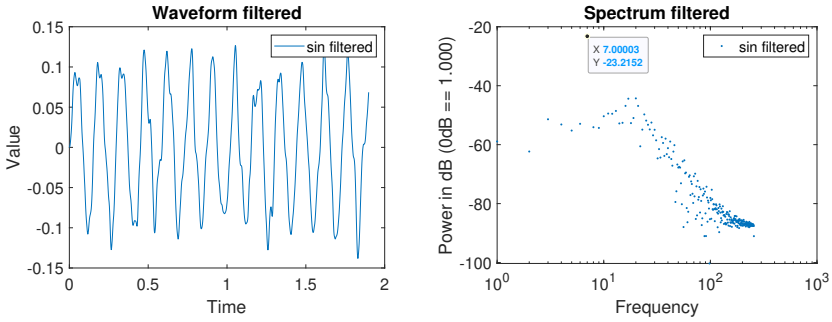
(a) Time diagram

(b) Frequency spectrum

Figure 13: Unfiltered $7Hz$ sine wave with additive Gaussian noise

After adding the same Butterworth filter as in the previous task which has $\omega_c = 2\pi \cdot 23rad/s$, its effect on the SNR can be evaluated. Figure 14 shows how the filter has influenced the signal.In Fig 15b, note how the filter has reduced the power of the noise above the cutoff

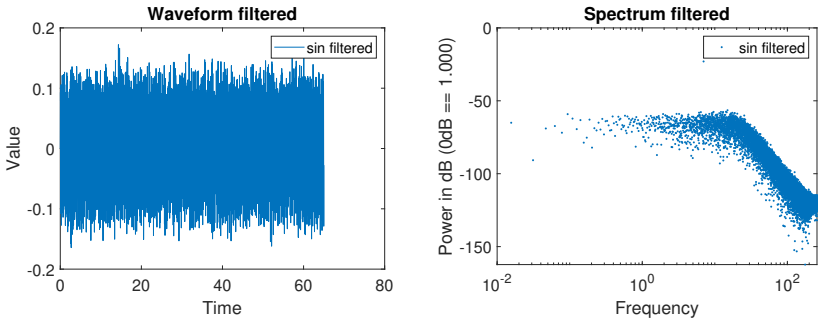
frequency, $f_c = 23Hz$, as it is expected to do. With the filter, the SNR is much better at 12.18 dB using Eq (6) due to the power of high-frequency noise being reduced.



(a) Sinusiodal with 7 hertz (b) The Fourier response of the sinus

Figure 14: Filtered sinus wave with additive Gaussian noise

Due to artifacts coming from the Fourier transform attempting to make the frequency spectrum periodic with a period equal to f_s and it not fully matching up at the edges due to the non-periodic high-frequency noise, increasing the simulation time will make for a better model as the effect of the artifacts is reduced. While the previous simulations in this section used a simulation time of 1.9s, the simulations were redone with a simulation time of 65s to see the effect of diminishing the effect of the artifacts. The new filtered output is shown in Fig 15.



(a) Time diagram (b) Frequency spectrum

Figure 15: Filtered sinus wave with additive Gaussian noise and 65s simulation time

The power of the sine wave is by and large unaffected at $-23.01dB$ since it is only the power of aperiodic signals, such as the Gaussian noise, which is reduced by increasing the simulation time as previously described. The noise floor, the maximum power of the noise, has therefore decreased, especially for high frequencies. At the highest frequencies, it decreased by $\approx 25dB$ while at the cut-off frequency, it decreased only $\approx 15dB$.

The SNR with the new simulation time of 65s is shown in Tab 2 with the SNR for some other simulation times for comparison. As the table shows, the improved accuracy in SNR has diminishing returns for very long simulation times. While the two 2dB reduction in SNR by going from 1.9s to 65s is considerable and therefore suggests a simulation time of 1.9s to be too short, going up to 1000s might end up taking a very long time for little improvement in SNR. While there is no reason to not go for a long simulation time as 1000s for such a simple model as in Fig 12 since its simplicity makes the simulation run fast anyway, for more detailed models keeping the simulation time to 65s will be enough. Judging by Tab 2, 65s seems to be a sweet spot. Reducing the simulation time to 1.9s on the other hand should except for in very extreme circumstances be avoided since a 2dB difference in SNR is considerable.

Simulation time	Unfiltered (dB)	Filtered (dB)
1.9s	-0.58	12.18
65s	-0.057	10.26
1000s	0.0076	10.24

Table 1: SNR for different simulation times

1.5 Conclusion

In this lab, the effects of sampling and its interactions with noise were studied in MATLAB and Simulink. We showed how frequency spectrum's in MATLAB will unavoidably include very low-power noise-like signals coming from arithmetic imperfections. It was also shown how aliasing can disguise itself as lower frequency signals which is why it is always important to filter out signals with a frequency above the bandwidth. With a low-pass filter like a Butterworth filter high-frequency noise can be removed to give a better SNR. Increasing the simulation time will give a more accurate SNR, though there are diminishing returns.

2 Lab 2: Variability and feedback

In this lab session, we explored the effects of parameter variability on an amplifier model using Cadence. The primary objective was to study how negative feedback counteracts circuit performance variation. Moreover, our exploration extended to how non-ideal gain in the operational amplifier, op-amp, can affect its function.

2.1 The amplifier model

A circuit model of the amplifier is shown in Fig 16. The input is driven by an AC sinusoidal voltage source. The amplifier which is controlled by a feedback loop increases the input voltage V_{in} by the closed-loop gain A_{CL} as a factor to make the output voltage V_{out}. This is shown in Eq (7).

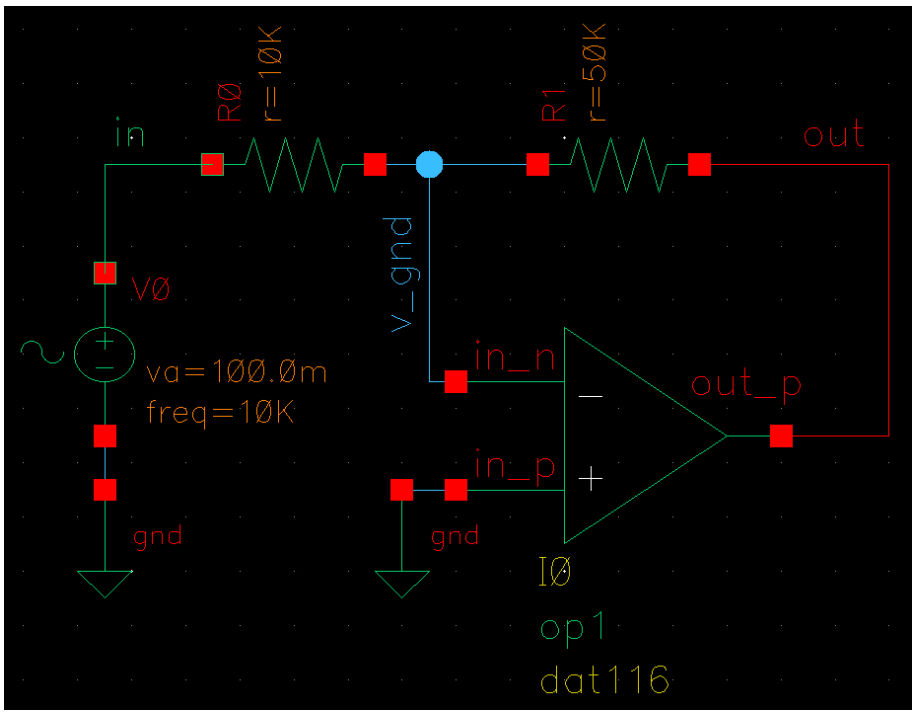


Figure 16: Schematic of the test circuit

$$V_{out} = A_{CL} V_{in} \tag{7}$$

An amplifier is, for this circuit model, largely controlled by the ratio, β, between the input and output resistor, R_{in} and the R_{fb}, as defined in Eq (8).

$$\beta = -\frac{R_{in}}{R_{fb}} \tag{8}$$

A_{cl} is calculated to be what is shown in Eq (9) were A is the gain of the op-amp.

$$A_{CL} = \frac{A}{1 + \beta A} \tag{9}$$

If the op-amp is ideal then $A \to \infty$ which makes A_{cl} as what is shown in Eq (10). It shows that A_{cl} depends more and more only on the ratio between the resistors with increasing A. This is then also the case for V_{out} which is shown in Eq (11).

$$A_{CL\infty} = \lim_{A \to \infty} A_{cl} = \frac{1}{\beta} \tag{10}$$

$$V_{out} = A_{CL\infty} V_{in} = \frac{V_{in}}{\beta} = -\frac{R_{fb}}{R_{in}} V_{in} \tag{11}$$

A simulation of the circuit model for an input amplitude of 100 mV for 1 ms, yielded the waveform graph in Fig 17. This corresponds with V_{out} from Eq (11) which when applied becomes Eq (12).

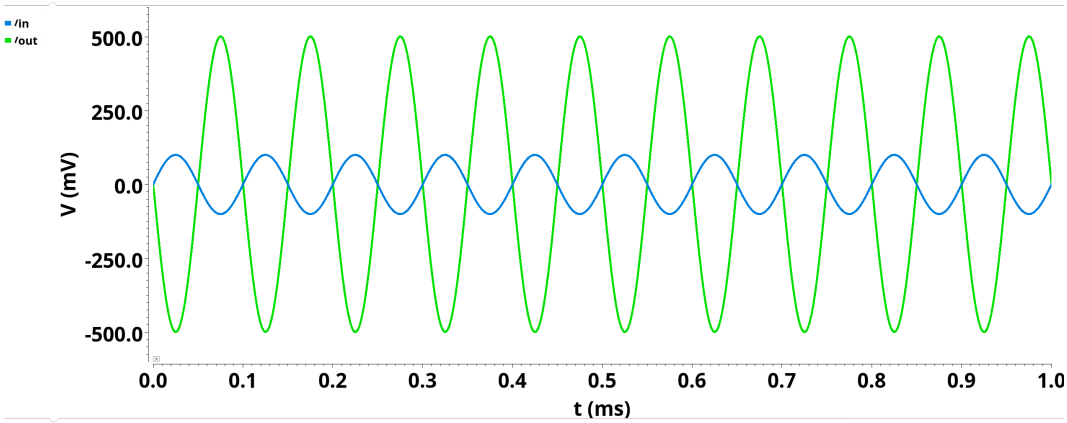


Figure 17: Transient response when $R_{in} = 10k\Omega$ and $R_{fb} = 50k\Omega$

$$0.5V = 0.1V \cdot \frac{50000}{10000} \tag{12}$$

2.2 Component variability

The variability of components can affect the characteristics of an amplifier greatly, though not always. To analyze the effect of a change in op-amp gain A for many different frequencies, the frequency response of the model circuit was measured and is displayed in Fig 18. From the figure, the cutoff frequency f_c was measured to be 1.48 Mhz, since frequencies above are dampened 3dB and more compared to lower frequency signals. Since the frequency response is of a low-pass filter, f_c is in this case also the frequency of the dominant pole of the circuit. The unity-gain frequency f_u, which is the point at which $V_{out} = V_{in}$, is at 6.81 MHz. //

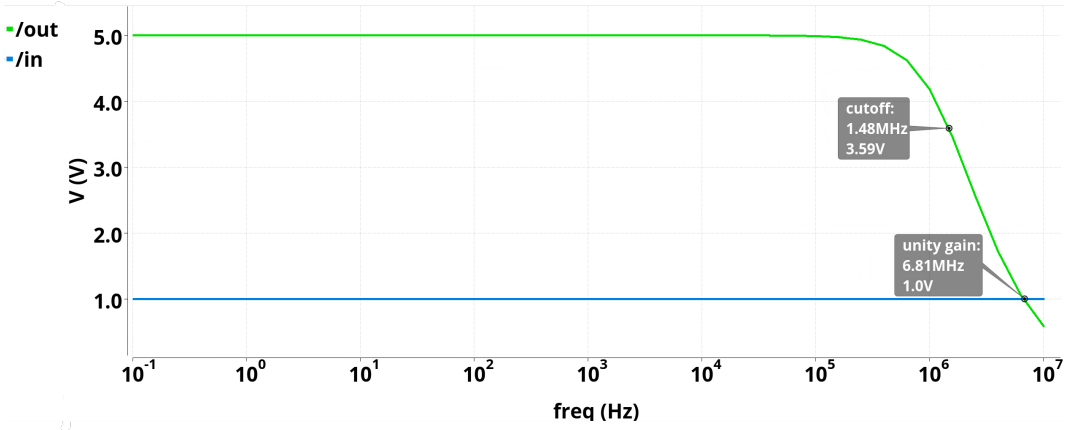


Figure 18: Frequency response when $R_{in} = 10k\Omega$ and $R_{fb} = 50k\Omega$

To get a representation of the frequency response of the circuit which is not dependent on the magnitude of V_{in}, a Bode plot can be made. It plots the gain of the amplifier in decibels depending on frequency. A Bode plot of the circuit model can be seen in Fig 19. It also shows the raw gain which is the gain from the virtual ground, the negative input to the op-amp in Fig 16, to the output. Fig 19 shows the same wide bandwidth as in Fig 18, since it is the same plot, but with a change in unit on the y-axis to decibel. The rate of decrease of the raw gain in Fig 19 is better expressed in terms of per decade since it is then constant across frequencies. The rate of decrease is then 20db/decade.

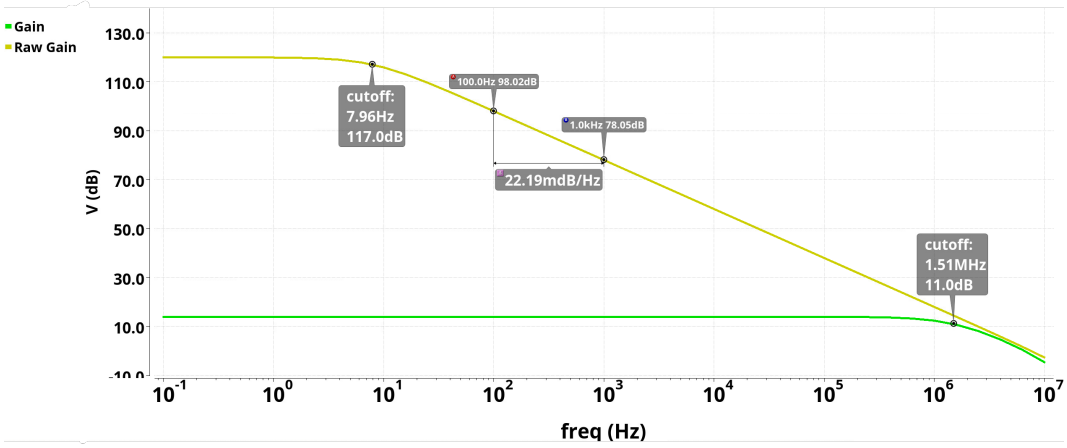


Figure 19: Bode plot the of the total amplifier gain and raw gain

To see how a change in A can affect the amplifier gain, the Bode plot in Fig 19 was appended with the gain for an amplifier with an op-amp with $A = 100$, compared to $A = 120$ as the amplifier in Fig 19 has. The appended figure is shown in Fig 20.

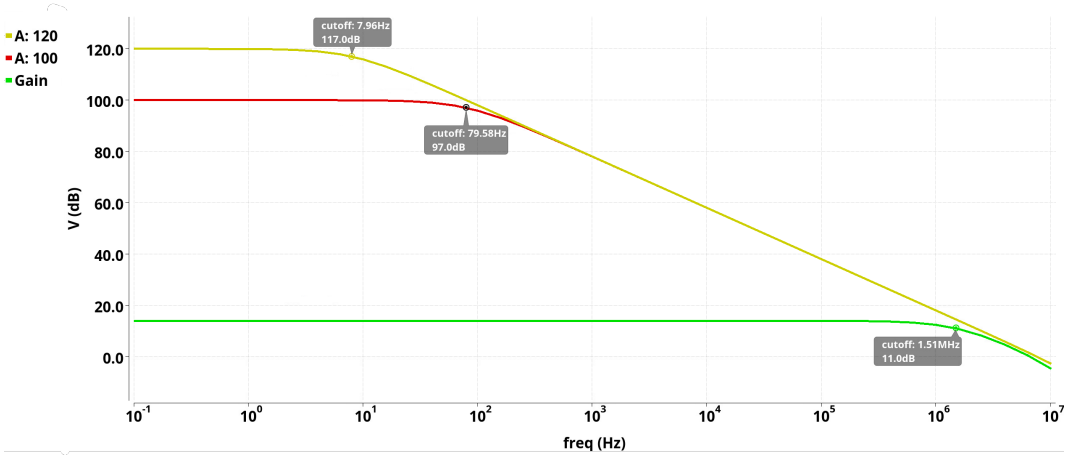


Figure 20: Bode plot of the amplifier gain for amplifiers with $A = 100$ and $A = 120$

Since the two amplifiers have the same gain no matter A for these A, the gain is plotted as one line in Fig 20. The raw gain is different, however. The figure shows how the raw gain starts at A and then decreases exponentially, since this is a logarithmic plot, once it reaches a certain f_c which depends on A. Since lower A has a higher bandwidth for the raw gain, the higher low-frequency gain of an amplifier with larger A does not make its raw gain higher than for an amplifier with lower A for frequencies above the f_c of the lower A amplifier.

2.3 Multi-component variability

By plotting the frequency response of the amplifier for many different values of a certain parameter, so-called sweeping, the frequency dependence of that parameter becomes a lot more deducible. Therefore we will in this section sweep over A, R_{in} and R_{out} to see their effect on gain over the frequency spectrum.

2.4 Single-parameter sweep

In this part, we are examining the effect of a changing A on the gain of the amplifier. Testing this as if appending the plots of more amplifiers with different A to Fig 20, A was swept from 20 to 120 with a step size of 10 to create Fig 21. It shows how even for low A the amplifier gain can be very similar to the case of a very high A such as when $A = 120$. A does not seem to affect the gain unless the raw gain, whose low-filter amplification is set by A, is lowered to the point of becoming close to the total gain of the amplifier.

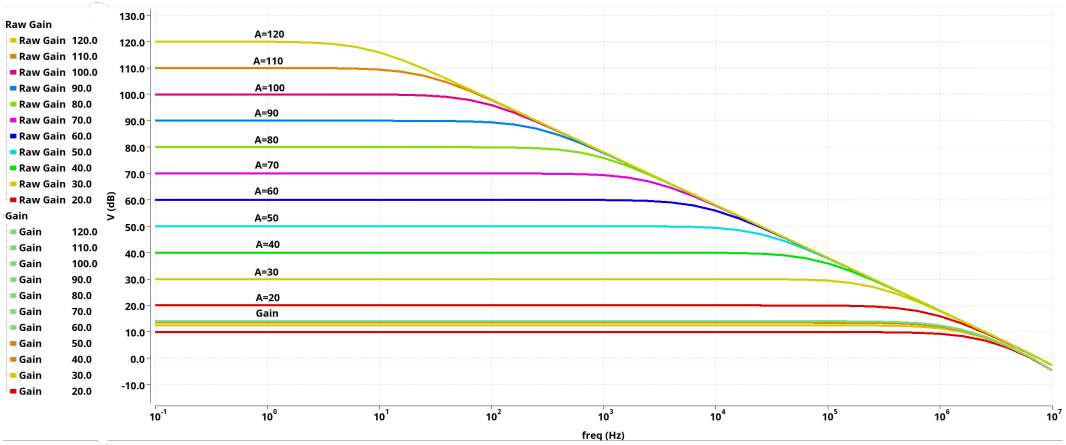


Figure 21: Bode plot of the amplifier gain for amplifiers from $A = 20$ to $A = 120$

The little effect A has on the output if it is large enough is even more visible in Fig 22 than in Fig 21. If $A \geq 50$, the difference is less than 2% of 5 V. This means A values in the neighborhood of around 100 are for this circuit pointless since a lower A is enough to make the op-amp behaves approximately ideal in this circuit.

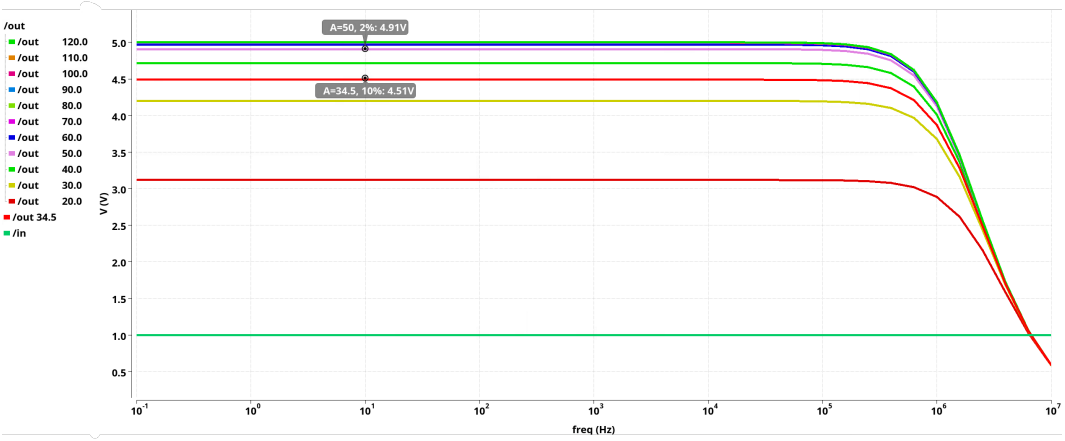


Figure 22: V_{out} for op-amp gain from $A = 20$ to $A = 120$

The performance of the amplifier decreases drastically however when $A < 50$. When $A = 34.5$ it is 10% from the ideal value and from then on it decreases even more rapidly. In looking at Fig 22, there very much seems to be a sweet spot of having $A \approx 60$ at

which the difference between the ideal and real value becomes negligible and making A higher becomes pointless. In a real-world application, however, the minimum possible A for the circuit can vary wildly. Real-world op-amps, as any active component, can have big parameter differences which means it is important to choose an op-amp type whose minimum A is large enough to not be a concern. It is therefore reasonable to have a typical A much higher than the previously mentioned sweet spot.

2.5 Multi-parameter sweep

The dependency of the amplifier gain on its resistor values can be checked by performing parameter sweeps on the values of R_{in} and R_{fb}. The sweep was done for each parameter in 10 steps in the range of $\pm 10\%$ its original value. A was set to $A = 120$. There are therefore in total 100 different test scenarios which are all shown in Fig 23. As expected, the figure shows V_{out} to be bigger the larger R_{fb} is in comparison to R_{in}, since the ratio between them decides $A_{CL\infty}$ which is all shown in Eq (11). The larger $A_{CL\infty}$, the higher the output will be when the op-amp is ideal. As could be seen in Fig **??**, the op-amp in this circuit is practically ideal when $A = 120$ which means V_{out} can be approximated as only dependent on the asymptotic gain.

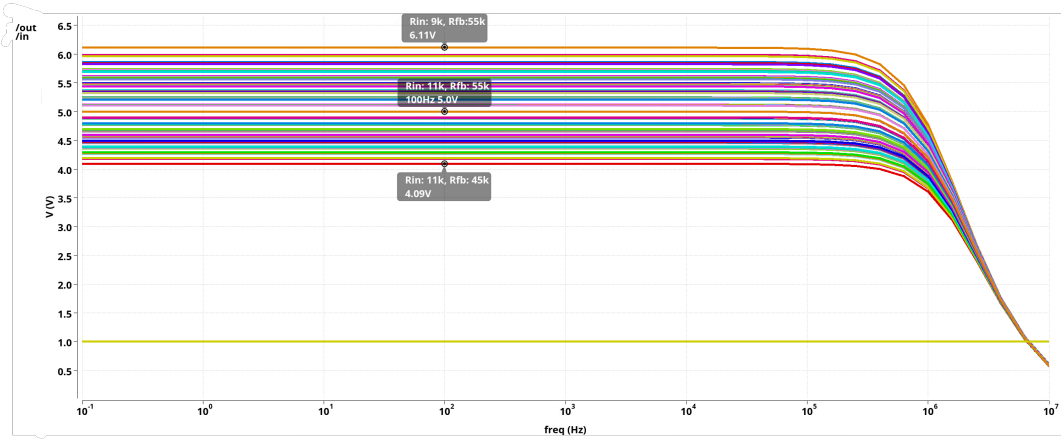


Figure 23: The effect of fully uncorrelated resistance variation on V_{out}

To further prove this, these calculations show the smallest and the largest amplification of the system

$$4.09V = 1 \cdot \frac{R_{fb} \cdot 0.9}{R_{in} \cdot 1.1} = 1 \cdot \frac{50000 \cdot 0.9}{10000 \cdot 1.1}$$

$$6.11V = 1 \cdot \frac{R_{fb} \cdot 1.1}{R_{in} \cdot 0.9} = 1 \cdot \frac{50000 \cdot 1.1}{10000 \cdot 0.9}$$

2.6 Parameter co-variation

Since a lot of the parameter error comes from a manufacturing gradient on the chip, i.e. in a certain area of the wafer the chips behave differently, the parameter values of adjacent components are often not completely uncorrelated as was assumed in Section 2.5. To see how assuming the error to be correlated affects V_{out}, the resistance values were multiplied with the factor r_{glob} which represents the parameter bias in an area of the wafer. The simulation in Section 2.5 was then appended with a sweep for r_glob from 0.9 to 1.1. The range for R_{fb} and R_{in} was also decreased to ± 2 %, Since most of the error is assumed to come from the manufacturing gradient. Each parameter is simulated at five uniformly placed levels in its range to not necessitate an excessive number of simulation runs. Since three parameters at five levels are being simulated, the total number of simulation runs becomes $5^3 = 125$. V_{out} for each simulation run can be seen plotted in Fig 24.

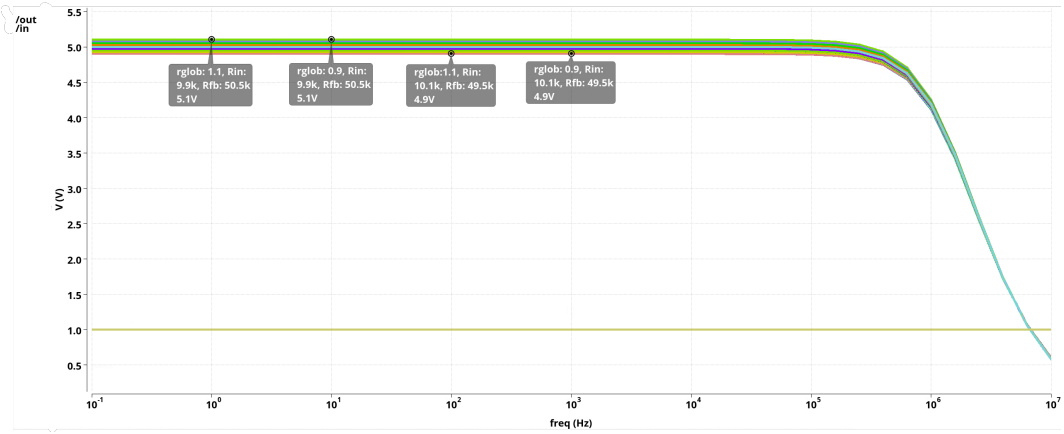


Figure 24: The effect of global resistance variations on V_{out}

From Fig 24 it is clear to see that the divergence from the ideal voltage of 5 V is much smaller than when assuming uncorrelated resistance values such as what is shown in Fig 23. The discrepancy is much smaller due to the asymptotic gain being decided by the ratio between R_{fb} and R_{in} as seen in Eq 11 and $A = 120$ making the op-amp behave ideally in this circumstance. Since the ratio stays the same if they are both scaled with the same factor, the value for rglob makes no difference on V_{out}. This is seen in the selected points in Fig 24 having the same R_{fb} and R_{in} but different rglob values and still having the same V_{out}. To find the maximum and minimum of these voltage discrepancies we calculated the higher and lower bounds as shown below. The calculated V_{out} extremes correspond with the minimum and maximum points shown in Fig 24.

$$5.1 = 1 \cdot \frac{50000 \cdot 1.02}{10000 \cdot 0.98}$$
$$4.9 = 1 \cdot \frac{50000 \cdot 0.98}{10000 \cdot 1.02}$$

2.7 Conclusions

In this lab, the focus was on conducting a comprehensive analysis of the variability of an amplifier using the EDA tool Cadence. This included showing how different kinds of variability can give very different outputs. We showed how as long as the op-amp gain A is large enough to make the raw gain be significantly higher than the overall gain, A does not matter since the op-amp can be assumed to be ideal. We also showed how the gain of the amplifier being controlled by the ratio between two resistors makes the amplifier immune from correlated changes in the resistances. However, uncorrelated changes can affect the amplifier significantly.

3 Lab 3

This lab is about different causes of noise and how to deal with them in A/D and D/A converters. With them, continuous signals can be converted into a set of discrete digital bits and vice versa. However, this conversion can significantly worsen the signal-to-noise ratio (SNR) which is why a solid understanding of a few causes of this is useful. For this reason, the effect of the choice of quantization and input power in an A/D converter will be analyzed. We will also look into two big causes of noise in the D/A converter. They are imperfect sample rate, jitter, and parameter errors which in this case is exemplified in an r2r ladder.

3.1 Uniform Quantization

For the first task, a simple A/D converter was made by using a one-dimensional lookup table in Simulink to map a sine wave input to discrete levels. This process is called quantization and the discrete levels are called quantization levels. The model can be seen in Fig 25 where the two upper boxes are scopes to be able to view the signals directly in Simulink. The simout box makes the output of the lookup table be stored as a workspace variable for further processing in MATLAB.

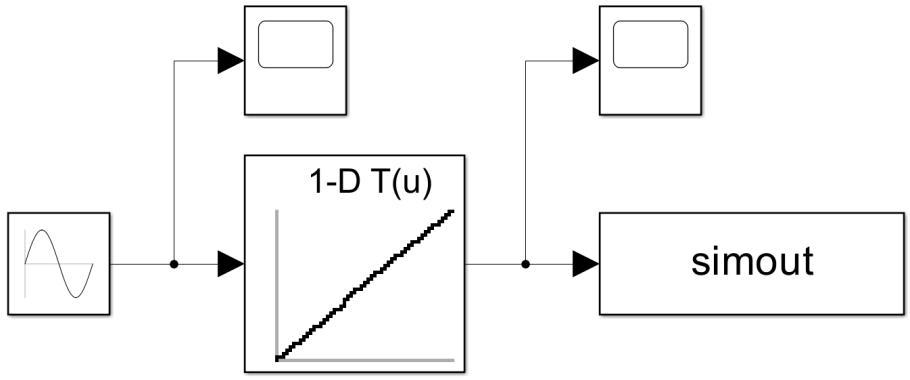
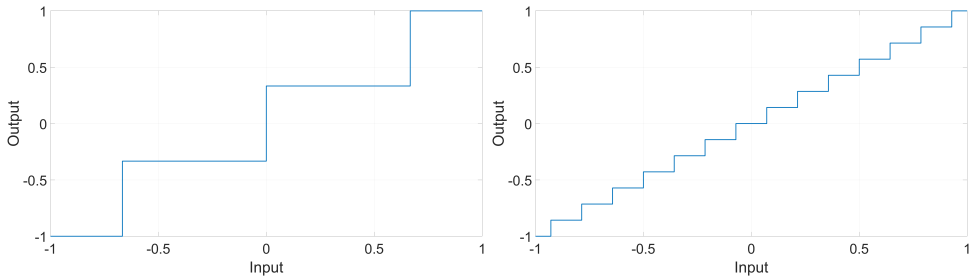


Figure 25: Simple A/D converter in Simulink

In Fig 26 two different uniform quantization can be seen. Since the mapping of an A/D converter works by moving the input value to the closest quantization level, an error will be generated for all of the cases where the input value does not happen to be exactly at a quantization level. The more quantization levels there are, however, the less rounding needs to almost always take place which means the mean error will for almost every input signal decrease. Therefore a quantization like in Fig 26b which has 15 levels will almost always have a smaller mean error than the quantization in Fig 26a which has 4 levels. The quantization functions shown in Fig 26 were generated with the **lutix** and **lutdata** Matlab functions which had the number of quantization levels as an argument.

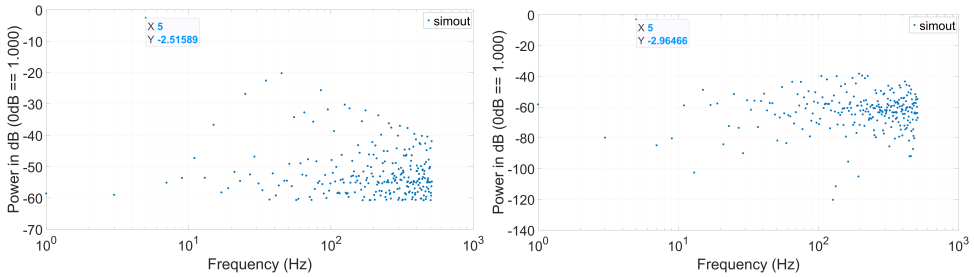


(a) 4 quantization levels

(b) 15 quantization levels

Figure 26: Uniform quantization

The amount of noise from the quantization decreases with more quantization levels as illustrated in Fig 27 by comparing Fig 27a and 27b. The error power of the 4 quantization level model is significantly stronger as seen in Fig 27a with the noise even reaching -20 dB compared to the 15 quantization levels in Fig 27b where a noise component frequency at most reaches -38 dB. This is because as previously discussed, the mean error for almost any input decreases the more quantization levels there are and the error is what is interpreted as noise. The input signal is for both cases a 5 Hz sine wave.



(a) 4 quantization levels (b) 15 quantization levels

Figure 27: Frequency spectrum of Uniform quantization

The point of quantization is to make a digital representation of a continuous signal by sampling it and storing the value of a point in some amount of bits. It therefore makes the most sense to sample to the point where there are as many quantization levels as there are numbers the bits can represent. Otherwise, accuracy could be improved for no increase in memory usage. When using the whole range of the bits, the number of quantization levels is then described by 2^N where N is the number of bits.

By sweeping from 3 to 14 bits used in the quantization of the model in Fig 25, the signal-to-noise ratio, SNR, could be calculated depending on the number of bits used. This is shown in Fig 28. It shows the SNR improving linearly by about 14dB/bit which is reasonable because with more quantization levels, the mean error is expected to decrease. It is the quantization error that causes all of the noise in Fig 28.

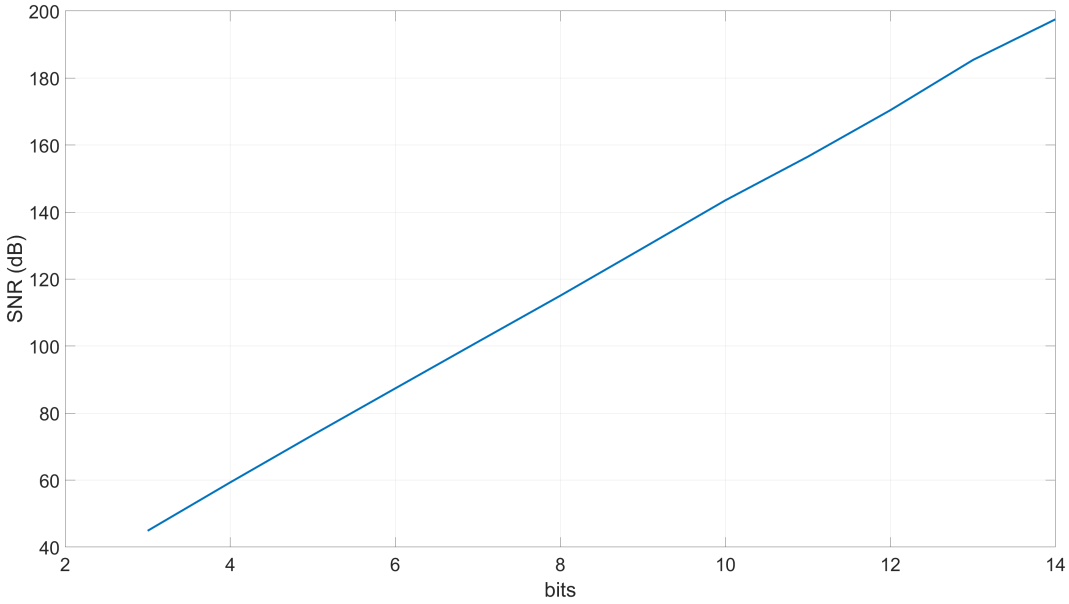


Figure 28: SNR per the number of bits used

In calculating the SNR, we assumed the quantization error can be likened to white noise which is not the case for all input signals. For example, if the input signal is a constant value, the error may for any number of bits used be constant since the same quantization level may

continue to be the closest. However, through this report, the input is a sine wave whose quantization noise can be approximated as white noise which means the aforementioned concern does not apply here. This is due to the sine wave input using a lot of quantization levels of a similar amount.

3.2 SNR vs input power

SNR can not only have a dependence on the number of bits used for the quantization but also on the input power of the signal entering the A/D converter. This is illustrated by Fig 29 for a 10-bit quantization. It shows how the SNR steadily increases with power to then suddenly collapses as the power goes past -3 dB.

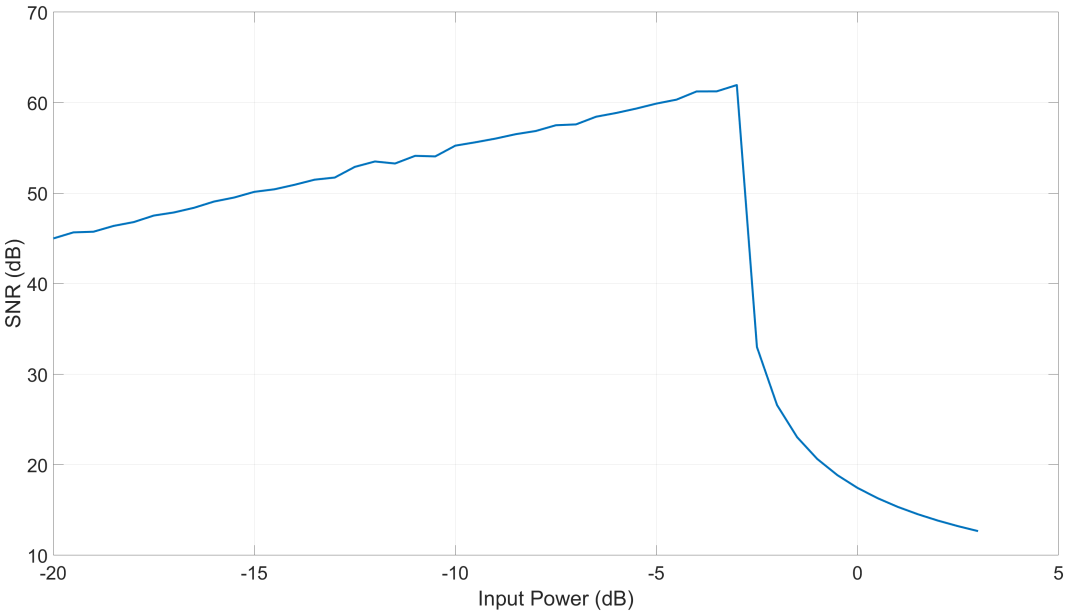


Figure 29: SNR for increasing input power

The improvement in SNR comes from the input utilizing more of the range of the quantization. Since Fig 29 is for a uniform quantization, the difference in-between each level is constant through the whole range. This means weak signals may only utilize a few of the quantization levels right around the bias point while a signal taking up the whole range will pass many levels and therefore have more information to more accurately reconstruct the signal from.

In Fig 29, the SNR collapses as the input power goes past -3 dB due to clipping. Since it is the amplitude of a signal that may cause clipping and not the power directly, it is first important to find the relationship in between. The average input power P_{avg} is a general characteristic of a signal which for a periodic signal $f(t)$ is defined by

$$P_{avg} = \frac{P}{T} = \frac{1}{T}\int_0^T |f(t)|^2 dt \tag{13}$$

where T is the period. For the case of a sine wave of amplitude A, $f(t) = Asin(\frac{2\pi t}{T})$, and P_{avg} then becomes

$$P_{avg} = \frac{1}{T}\int_0^T |Asin(\frac{2\pi t}{T})|^2 dt = \frac{A^2}{2T}\int_0^T (1-cos\frac{4\pi t}{T})dt = \frac{A^2}{2T}(T-[\frac{Tsin\frac{4\pi t}{T}}{8\pi}]_0^T) = \frac{TA^2}{2T} = \frac{A^2}{2} \tag{14}$$

For example, the average power of a sine wave when $A = 1$ is then $P_{avg} = \frac{1}{2}$. From Eq (14), A can be singled out as

$$A = \sqrt{2P_{avg}} \tag{15}$$

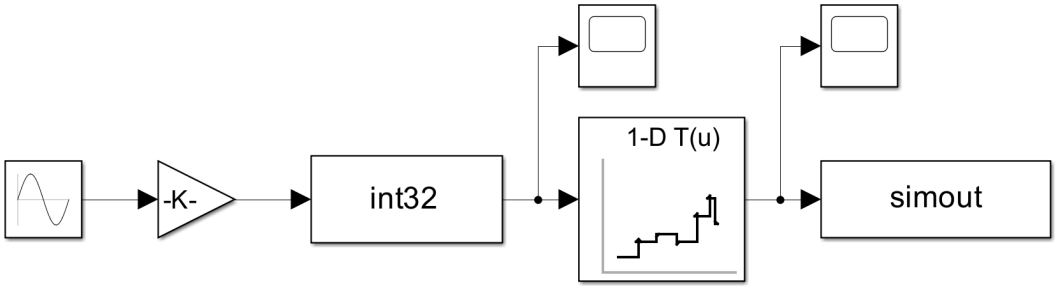


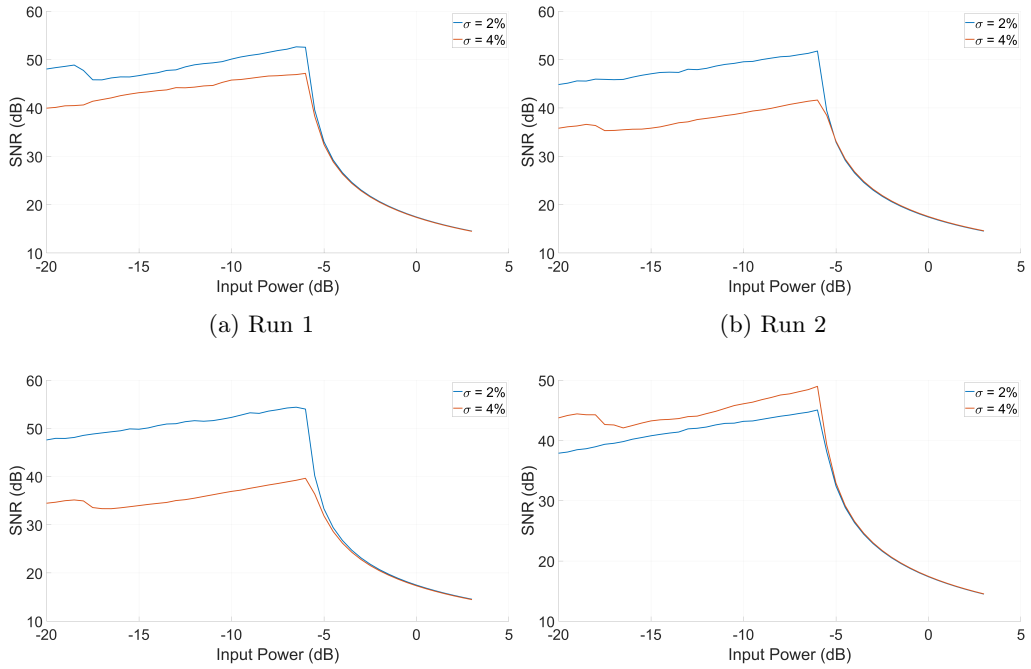
Figure 30: Simulink Model of a D/A converter

An input power of -3 dB means the signal has $\frac{1}{2}$ unit power which corresponds to a sine wave having an amplitude of 1 according to Eq (15). Since the lookup table standing in as a D/A converter has an input range of ± 1, the signal will be capped to within that range for the D/A conversion. An input power of -3 dB is therefore the max power of a sine wave before clipping occurs. The sharp decrease in SNR comes from the clipping error making the digitized representation correspond less to the continuous signal. The difference is interpreted as noise which reduces the SNR.

3.3 Non-uniform Quantization

A digital-to-analog (D/A) converter can also be modeled with a lookup table as shown by Fig 30. It works by the amplifier (illustrated by a triangle) expanding the input sine wave to have a maximum magnitude equal to the maximum of the digital representation. Then the current signal value is equal to what it would have been if was a digital value represented in bits, if only it were rounded to each such level. That rounding is what the look-up table first does to then with an r2r ladder map the digital representation to a value for the output signal. With an interpolation algorithm, it then tries to reconstruct the original sine wave. The output is sent to the MATLAB workspace as the variable simout.

Any converter will have imperfections however, as is illustrated by four runs of r2r ladders with either $\sigma = 2\%$ or $\sigma = 4\%$ were σ is the standard deviation error of each resistor in Fig 31. This means the errors in each r2r ladder are random and different r2r ladders are expected to have different characteristics. As can be seen in Fig 31, the SNR of the converter can vary wildly, especially when $\sigma = 4\%$, as the larger σ means the errors which compound are larger. How big the errors are can vary wildly depending on random chance. This means it is even possible for the r2r ladder with $\sigma = 4$ to have a smaller error than when $\sigma = 2$ as seen in Fig 31d though that is generally not the case. Due to the compounding effect a doubling in σ has going from $\sigma = 2\%$ to $\sigma = 4\%$ often leads to a much larger error than just doubling as Fig 31 show. Each r2r ladder contains $2N$ resistors which means the 10-bit D/A converter in Fig 31 has $2 \cdot 10 = 20$ resistors in its r2r ladder. There are therefore plenty of resistors for the errors to compound.



(a) Run 1

(b) Run 2

(c) Run 3

(d) Run 4

Figure 31: The SNR for 4 simulation with different r2r ladders having $\sigma = 2\%$ or $\sigma = 4\%$ error

3.4 Sample Jitter

In theoretical models of D/A and A/D converters, the sampling rate is assumed to be one constant frequency. This means every consecutive sample should in theory have the same fixed time interval in between. In the real world, however, the sampling will not happen perfectly but will instead have small deviations from its true periodicity which are called jitter. To simulate the effects of jitter a model shown in Fig 32 was made. It adds a random deviation of up to 10% of the derivative of the signal to the original. This essentially introduces a random deviation to the timing of the samples. The error appears as noise which is similar to the timing variations affecting a converter in a real-world scenario.

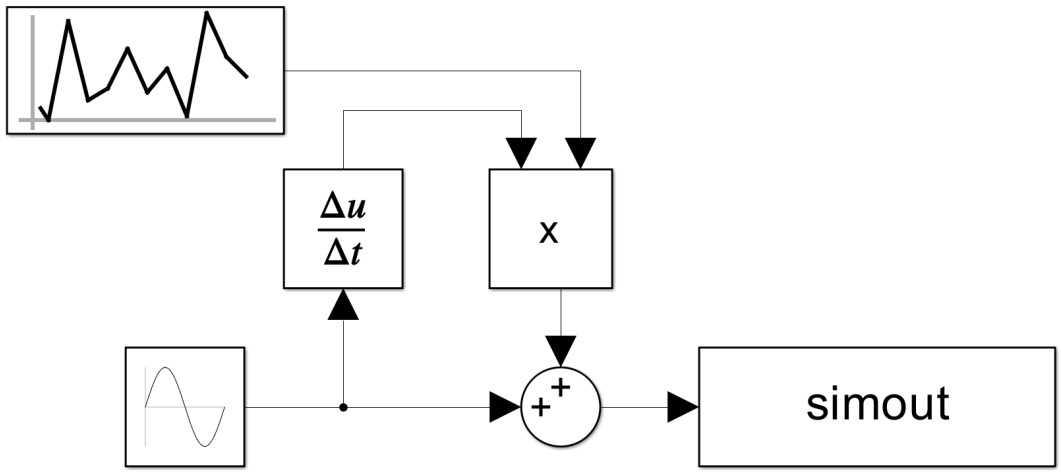


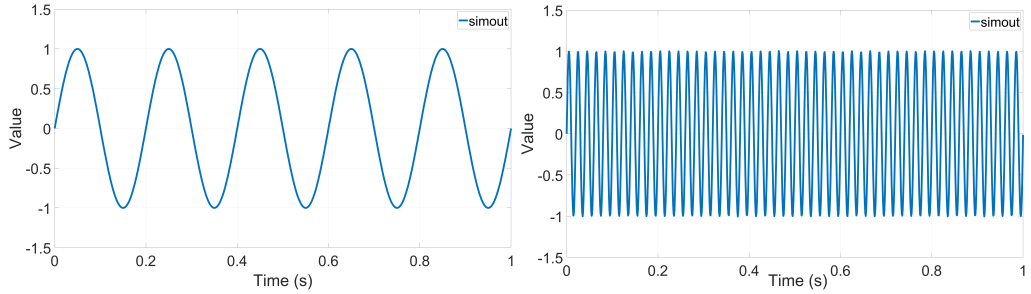
Figure 32: Simulink model to simulate the effect of jitter on a sine wave

Fig 33 show the time diagram of two different sine waves with $f = 5\,\text{Hz}$ and $f = 50\,\text{Hz}$ which have had jitter applied to them. The effects of jitter are not very noticeable, especially

for Fig 33a though in the frequency spectrum of the sine waves which are shown in Fig 34 it can be observed that the strength of the noise caused by jitter has increased by approximately 20 dB by going from 5 Hz to 50 Hz. A 20 dB increase corresponds to a 100 times increase in power which is reasonable because the jitter power P_{err}, not P_{avg}, corresponds to signal frequency according to

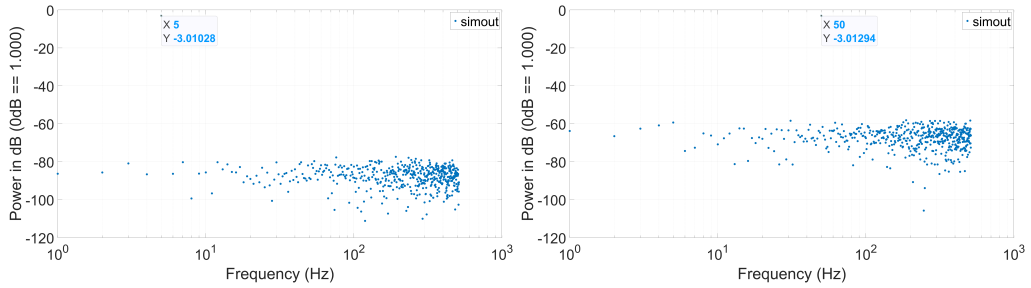
$$P_{err} \propto f^2 \tag{16}$$

and the frequency has increased by a factor of ten by going to 50 Hz. The noise caused by jitter is therefore considerably more powerful the faster the sampling rate is. The increased jitter power is reflected in considerably worse SNR as shown in table 2.



(a) $f = 5Hz$ (b) $f = 50Hz$

Figure 33: Input signal in time



(a) $f = 5Hz$ (b) $f = 50Hz$

Figure 34: Noise caused by jitter for two sine waves

Frequency (Hz)	SNR (dB)
5	55.4
50	35.2

Table 2: SNR for sine wave with only jitter noise

The jitter power does not only have a dependence on the frequency but also on input signal power. To examine this dependence a sweep over the input power from -40 dB to 3 dB was done for a 50 Hz sine wave. As can be seen in Fig 35, both signal and jitter noise power increase with input power. Since in this model all noise comes from the jitter, the SNR shown in Fig 36 is the ratio between the signal and jitter error power. The SNR is constant because as seen in Fig 35 both signal and jitter power depend equally on input power. This is because input power increases the amplitude according to Eq (15) and both are proportional to A^2 as is also shown by the powers growing linearly in a decibel plot. The constant SNR is to be expected as the relative error caused by the jitter is the same no matter the amplitude.

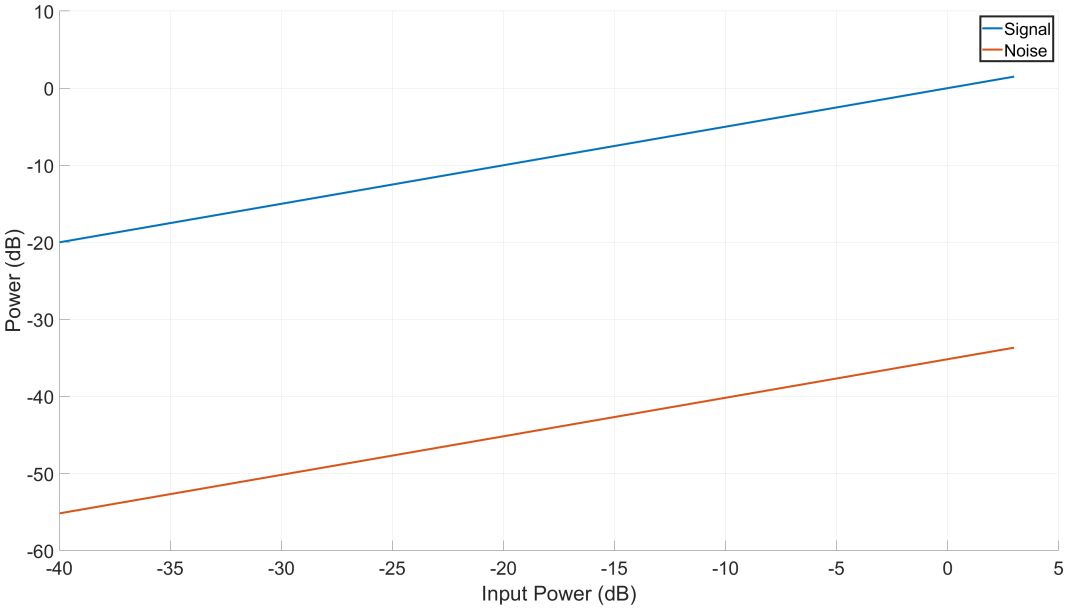


Figure 35: Output power for increasing input power

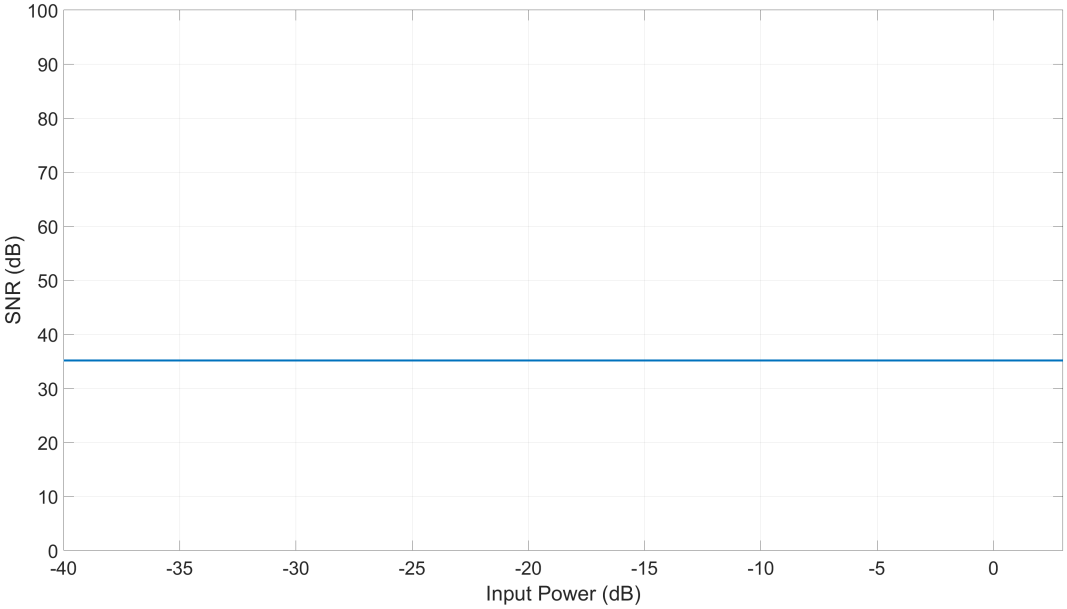


Figure 36: SNR for increasing power with jitter causing noise

3.5 Conclusion

To summarize, in this lab various aspects of D/A and A/D converter were explored. We showed that the number of quantization levels significantly affects the accuracy of the conversions. A/D converter accuracy also depends on input power whereas higher power signals have better accuracy as they cross more quantization levels unless they grow so large so at be outside the input bounds of the A/D converter. Then the SNR will plummet as the converter will sample a clipped input signal. In a D/A converter, small imperfections in its components compound worsen SNR significantly though this can vary a lot due to the randomness of such errors. Periodicity imperfections in the sample rate, so-called jitter, have been shown to cause errors that increase strongly in power with increasing frequency. Increasing input power does not worsen the jitter.

4 Lab 4

In order to remove unwanted frequencies which can be aliased into the base band, it is important to be able to construct an anti-aliasing filter which can attenuate such high-frequency components effectively. In this lab we will show how one such filter can be designed by implementing poles which describe a suitable transfer function as a cascade of Sallen-Key blocks. We will also show how the characteristics of these filters can be affected by variability.

4.1 Filter design

The filter to be designed is a lowpass filter confined to the area outlined in Fig 37. This

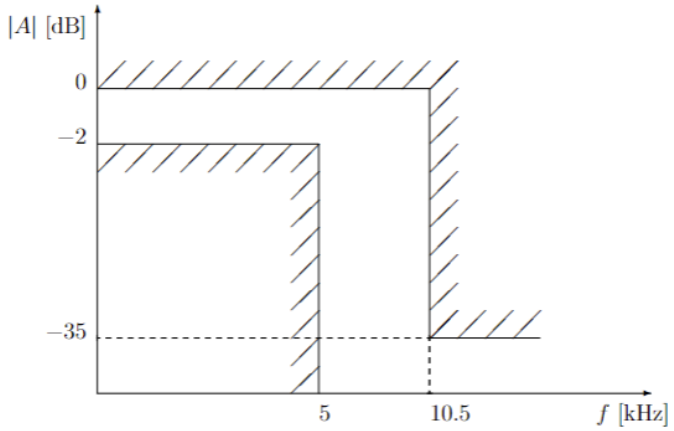


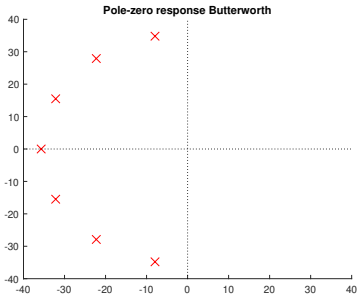
Figure 37: Graphical representation of filter specification

means the required specifications are:

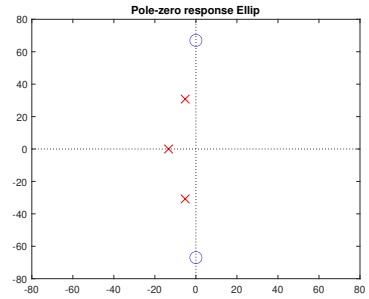
- Cutoff frequency: 5 kHz
- A maximum of 2 dB ripple
- Attenuation of 35 dB from 5 kHz to 10.5 kHz

As for filter types that are supposed candidates which are Butterworth, Chebychev I and II and an elliptical filter. To decide which filter to use we are looking at two criteria, first the transfer-function should have no zeros (except at infinity) and secondly it should allow lowest-order implementation possible. To account for filter variability there is to be a 4 % margin.

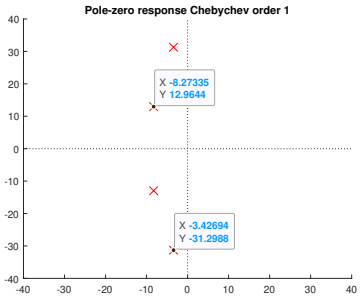
Calculating each in Matlab gives us four different filters where the orders are 6,4,4 and 3. The Chebychev filters are in order 4 and the elliptical filter is of order 3. To further inspect these we can observe the Fig 38 in this figure we see that zeros are represented by a circle and the poles are represented as a cross.



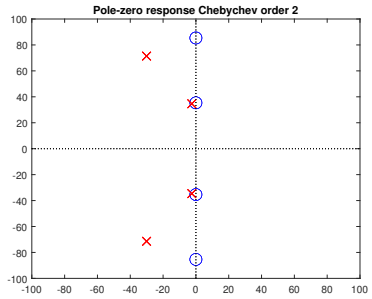
(a) Butterworth has a pole-zero response with seven poles



(b) Elliptic has a pole-zero response with three poles



(c) Chebychev I pole-zero response with four poles



(d) Chebychev II pole-zero response with four poles

Figure 38: Filtered sinus wave with addative gaussian noise

From the figures above one can conclude that the only good representation of the filter is a Chebychev of order 4 that's the lowest possible order filter of these four while at the same time having no zeros available. The resulting transfer function for this attained filter is depicted in Eq 17.

$$\frac{1.8625 \cdot 10^{17}}{(s^2 + 1.655 \cdot 10^4 s + 2.365 \cdot 10^8)(s^2 + 6854s + 9.914 \cdot 10^8)} \tag{17}$$

Inspecting the transfer function frequency response the graph in Fig 39 was attained. Here one can visibly see how the ripple is in tune with our restriction zero to -2 dB, further from 5 kHz to 10 kHz we can observe a drop in magnitude of about 37 dB.

4.2 Implementing the filter

When implementing the filter in the EDA tool Cadence, a Sallen-Key(SK) filter was chosen as a setup seen in Fig 4.2. In this setup, the filter is configured by two resistors called R_1 and R_2 and two capacitance's called C_2 and C_1. The capacitors have been set to strictly be 10 nF for this instance, whilst the two resistors are the configuration as well as the necessary feedback.

The components R_1 and R_2 and the amplification A can be calculated with the following Equations where P_1 and P_2 are the pole pair combinations calculated in the filter design part.

$$R = \frac{1}{C} \cdot \frac{1}{\sqrt{p_1 \cdot p_2}} \tag{18}$$

$$A = 3 + \frac{p_1 + p_2}{\sqrt{p_1 \cdot p_2}} \tag{19}$$

Computing this we attain that R_1 and R_2 are $6.5025 \cdot 10^3$ and $3.1760 \cdot 10^3$ with the amplification A 1.9241 and 2.7823 respectively.

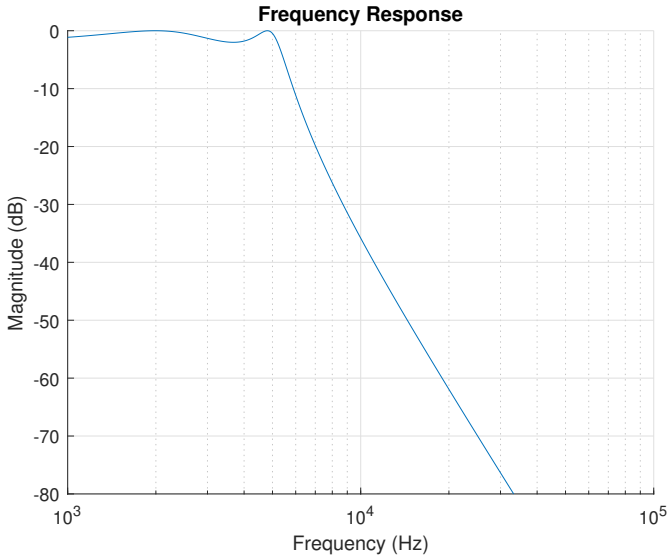


Figure 39: Caption

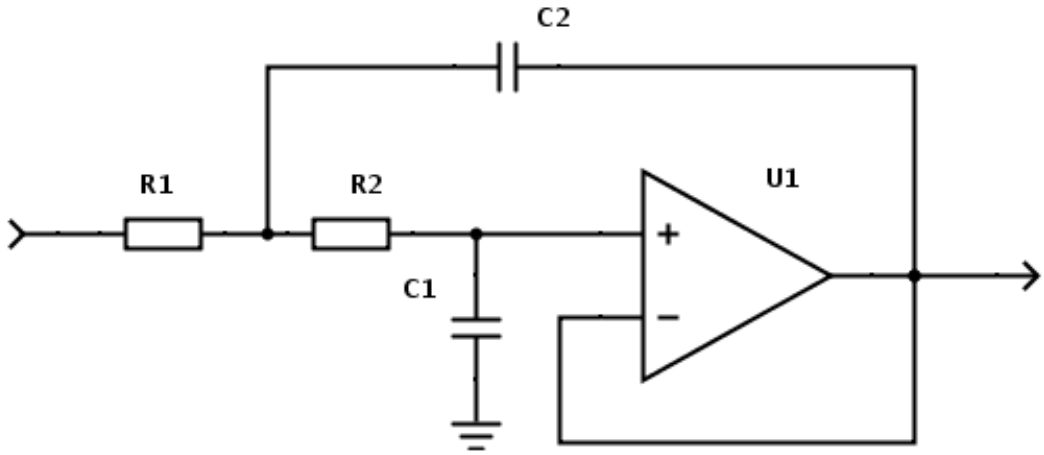


Figure 40: Circuit schematic of one Sallen-Key block

4.3 Circuit Implementation

The filter will be made in a cascade of two Sallen-Key blocks each contributing a pole pair. The Sallen-Key blocks have two parameters from which the conjugate pole pair is decided. They are the closed-loop gain of the amplifier inside and the resistance rbp which gives the resistance value for two resistors leading up to the amplifier. There are also two capacitance's which are set at a constant 10 nF.

By sweeping over the parameters of a Sallen-Key block their dependence will become visible. This is shown in Fig 41. It shows how a lower rbp delays the response of the filter to start at higher frequencies, especially when rbp is 1 kΩ. This is presumably because the jump from 6 kΩ to 1 kΩ is relatively the largest one. For the case of a gain of 3, large spikes appear which are delayed in accordance with the strength of rbp. This is because as was deduced in the exercise session,

$$p_1 + p_2 = \frac{A-3}{RC} \tag{20}$$

By setting the parameters of the Sallen-Key blocks in accordance with what was calculated in Section 4.2, the filter was implemented as shown in Fig 42. Since the implementation

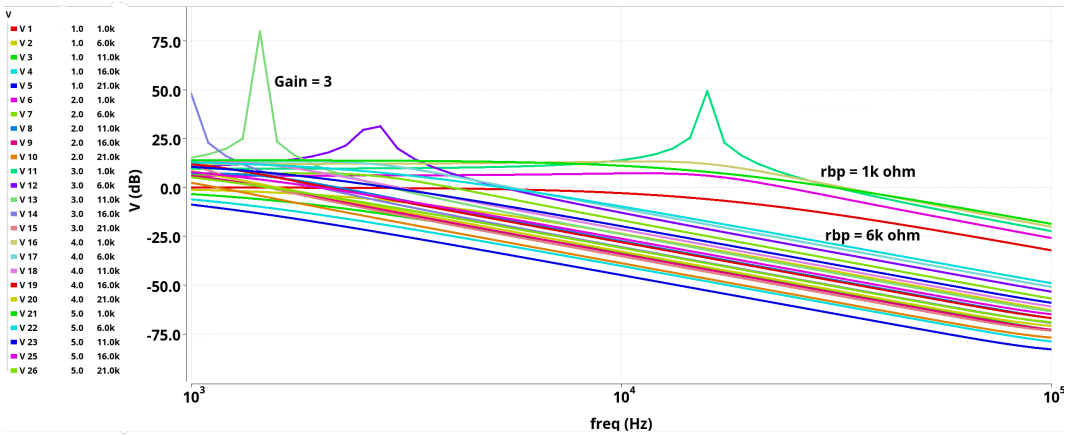


Figure 41: Response of Sallen-Key depending on gain (left) and rbp (right)

of the filters requires the blocks to have a gain above one, a resistance divider is set by the output to make the low-frequency gain only slightly less than one. The frequency response of the filter is as shown in Fig 43 where the passband and stopband corners for which the filter has been optimized have been marked. It shows the filter working correctly for ideal components.

4.4 Component Inaccuracies

Since the filter function depends on its parameters which are subject to error, it is worth analyzing how errors can prevent the filter from working as intended. To ease calculations, all errors are approximated as either random errors which are not locally correlated and global errors which affect all components in the filter equally. The parameters affected are those of the capacitances and the resistances both in the feedback circuit and preceding the amplifier as seen in Fig 40.

Table 3 shows how many of the 100 generated filters generated in Monte Carlo simulations did not pass the passband and stopband corners successfully. The inverse of this is the yield which shows how the yield becomes considerably worse when both mismatch sources are combined. While Table 3 shows the global errors to lead to considerably more invalid filters than for random errors this does not necessarily mean global errors are worse since for these simulations the global variations in capacitance were chosen to be larger than the fully random variations.

Source	Invalid (%)
Random	16
Global	32
Both	45

Table 3: Invalid filters from 100 randomly generated filters according to source

Fig 44 shows how random variations do not move the filter frequency response, but rather change the strength of the ripple around the passband corner. Since all of the filters have had their response attenuated enough to not violate the stopband corner, all violations happened at the passband corner. The reason why the strength of the ripple changes is because of the Q-factor, which is the ratio between the imaginary $X(\omega)$ and real $R(\omega)$ part of the poles as shown by Eq (21), is most likely to change as the changes in $X(\omega)$ will largely be uncorrelated with $R(\omega)$ as they are dependent on uncorrelated changes in parameter values. It is the Q-factor that decides the strength of the ripple. The cutoff frequency does not seem to change however which is because it is unlikely for the errors in many of the parameters

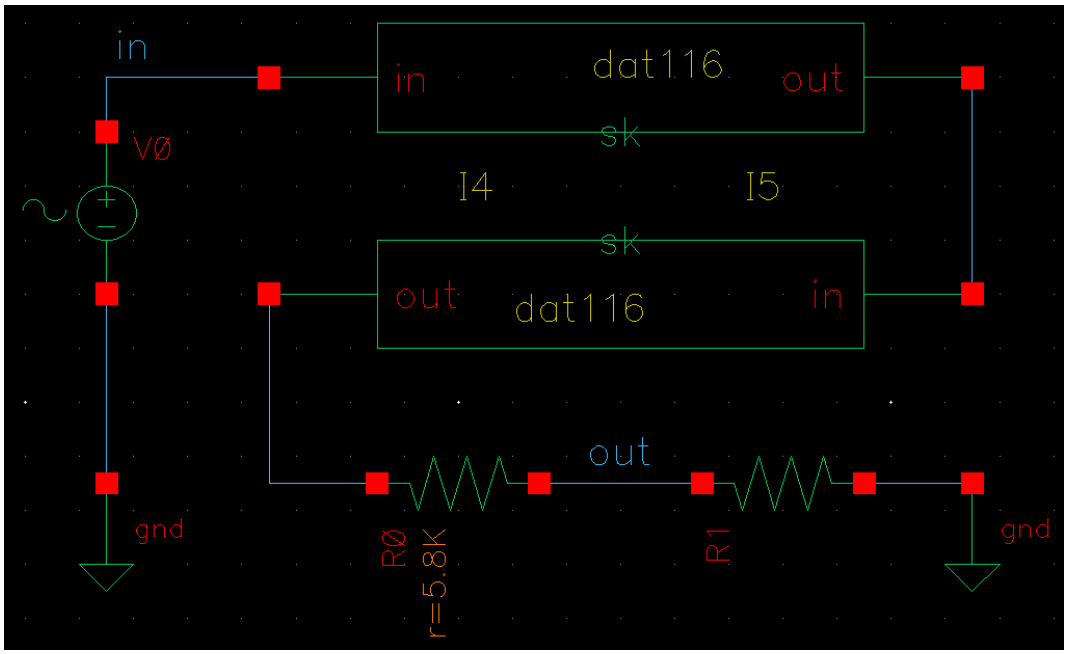


Figure 42: Circuit schematic of the filter with Sallen-Key blocks

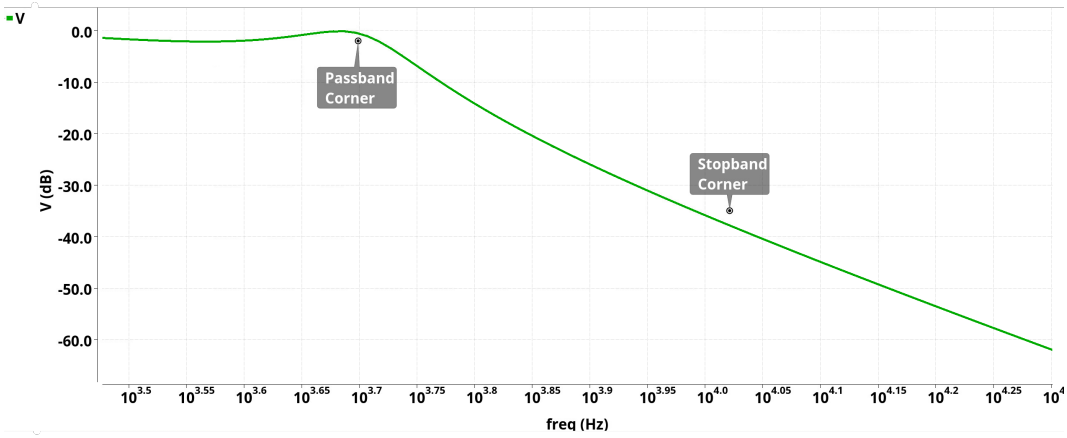


Figure 43: Filter response for frequencies between 3 kHz to 20 kHz

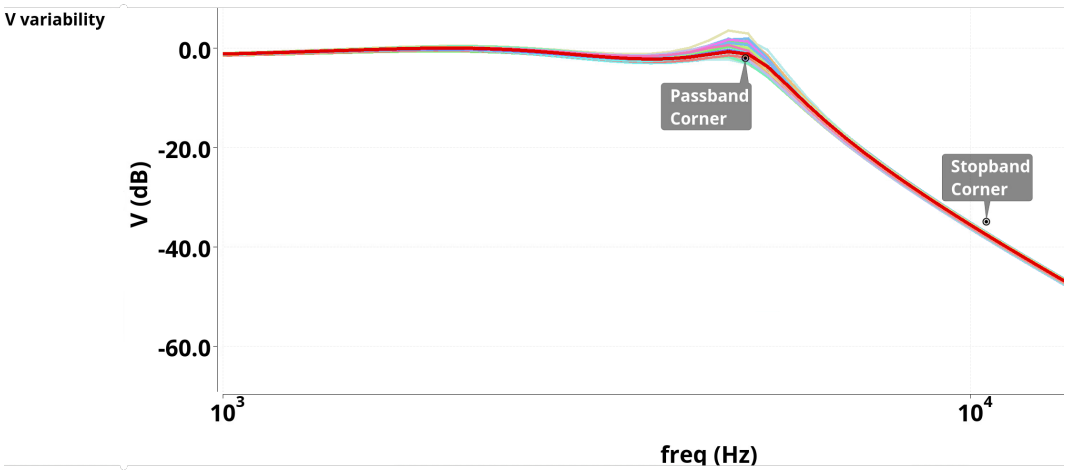


Figure 44: Filter response for only random parameter variability

to correlate enough to move a pole pair. Pole pair placement is decided by an RC product, as seen in Eq (22), whose changes in the terms are not correlated. Since only the Q-factor is likely to change much, it, therefore makes sense for only the passband corner to have been violated for random variations as it is by the ripple, whose strength is decided by the Q-factor, is.

$$Q = \frac{X(\omega)}{R(\omega)}, H(j\omega) = \frac{1}{R(\omega) + jX(\omega)} \tag{21}$$

$$p_1 p_2 = \frac{1}{R^2 C^2} \tag{22}$$

Fig 45 shows how global variations only move the frequency response to happen at lower and especially higher frequencies while the size of the ripple stays the same. About half of the violations came from violating each of the corners in this case. The frequency response having moved comes from the change of the RC factor for all pole pairs being constant which means all poles have the same relative shift as suggested by Eq (22) which is the equation for one Sallen-Key block. In the s-plane, this would show up as all poles moving in the same direction to the same relative degree. It will therefore only be the magnitude of the poles which will change which moves the cutoff frequency. Since the Q-factor will stay constant as all parameters change the same relative to each other, the strength of the ripple will stay the same.

Fig 46 shows the sum of both error sources shown in Fig 44 and 45. As expected it therefore has the most corners violated as both the strength of the ripple changes as well as the response itself has been moved.

4.5 Conclusion

In this report, we showed how a low-pass filter can be made according to certain specifications. By comparing the needed poles to create different types of filters, a Chebychev I filter was chosen as it had the lowest order and did not require zeros. This filter was then implemented by putting two Sallen-Key blocks in a cascade and setting their gain and some resistances to specific values to make them function as one pole pair each. We lastly showed how uncorrelated errors increase the ripple of the filter, while correlated errors can either move ahead or delay the frequency response of a filter.

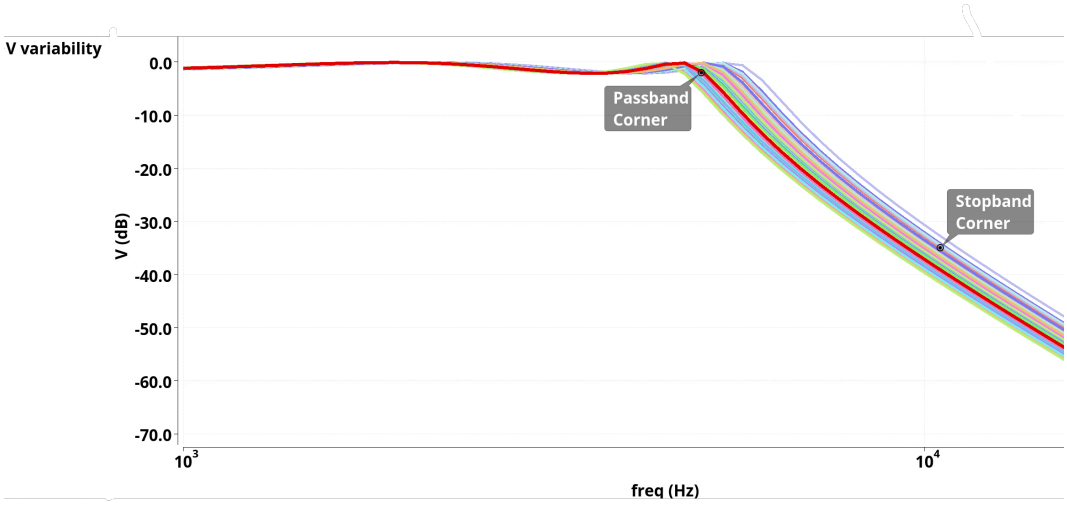


Figure 45: Filter response for only global parameter variability

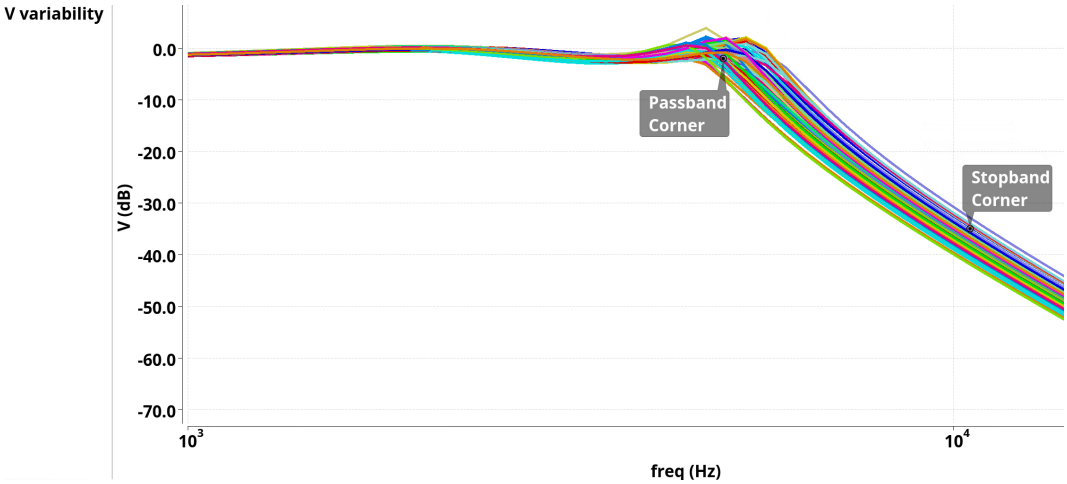


Figure 46: Filter response with both parameter variability sources

5 Lab 5

This report will be about how nonlinearities in continuous-time circuits are handled and how they can act in a circuit. To do this the report uses cadence to produce a circuit analysis and then post-process the data by transferring it to Matlab.

5.1 Cadence-MATLAB data transfer

In this simulation the same circuit will be used as in lab two which can be seen once again in Fig 47 where the schematic consists of an op-amp which is described in further detail in lab 2.

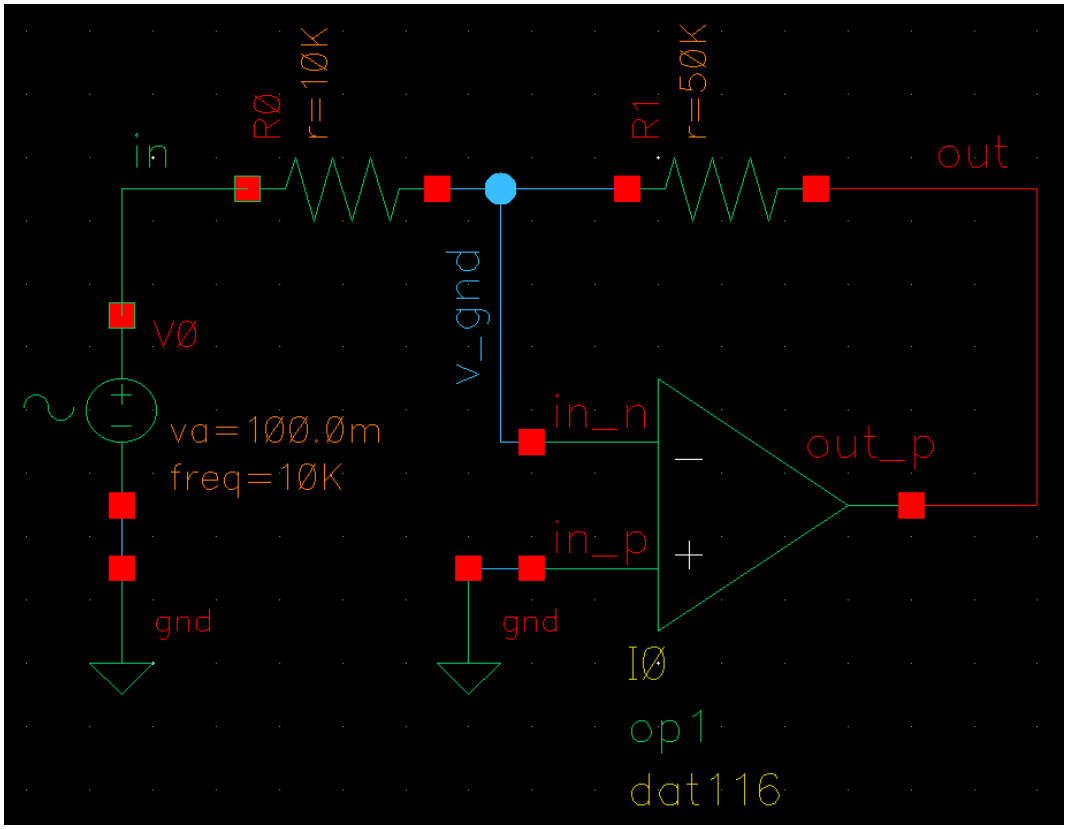


Figure 47: Circuit schematic for the test circuit

When simulating this circuit, the output is an XML file that encapsulates all essential elements of the simulation. This XML file serves as input data for further processing in Matlab.

Plotting the simulation waveform the Fig 62 as well as its spectra form for ≈1000 data points.

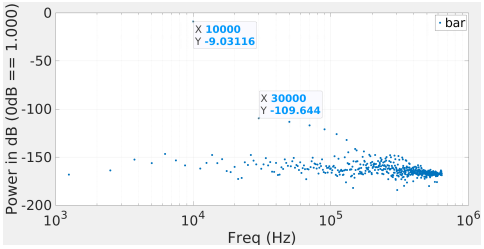


Figure 48: Spectrum with 1024 data points

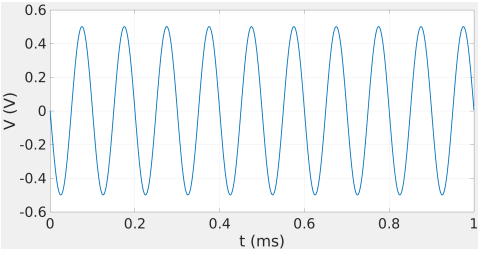


Figure 49: Waveform chart of the circuit

with DC?	≈1000 steps (dB)	≈15000 steps (dB)
Yes	51.6	51.6
No	91.3	161.0

Table 4: SNR for different step sizes and with or without the 0 Hz noise

By increasing the number of data points the noise floor gets lower as can be seen in Fig 50 were bar2 is for ≈15000 data points instead of ≈1000 data points. This however comes at the cost of increased simulation time which went from 22.6 ms to 228.8 ms.

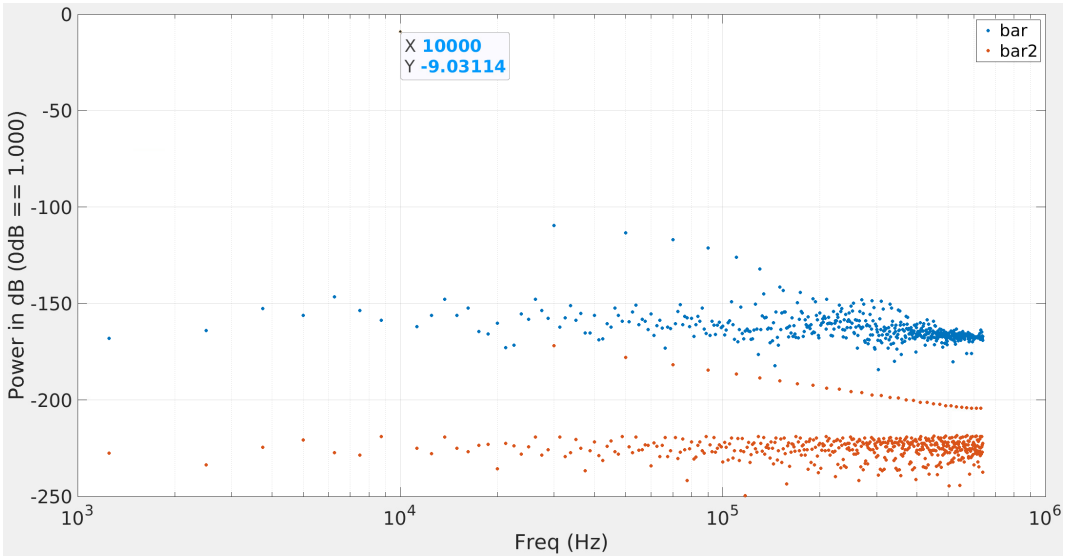


Figure 50: Frequency spectrum for ≈1000 points and ≈15000 points

In Fig 50 there is also a significant DC component which is not shown. As there is no DC input the DC-component is noise. As can be seen in Table 4 the noise caused by the DC component is so much larger than the other noise frequencies, that increasing the number of steps does not change the SNR. This is because the DC error is not a quantization error, but an intrinsic error of the amplifier. Since the DC component is easily distinguishable in the frequency spectrum, it is easy to remove it digitally. The improved SNR is also shown in Fig 4.

5.2 SNR as function of input level

This next part consists of performing a sweep on the test circuit with a range from 1mV to 10V to see the impact of voltage for the circuit. Conducting a sweep with twenty logarithmic steps each we get the frequency spectrum shown in Fig 51. In the figure, the legends bar3, 1 - 20 are concurrent with the amplitude meaning the higher the number the higher the input voltage is.

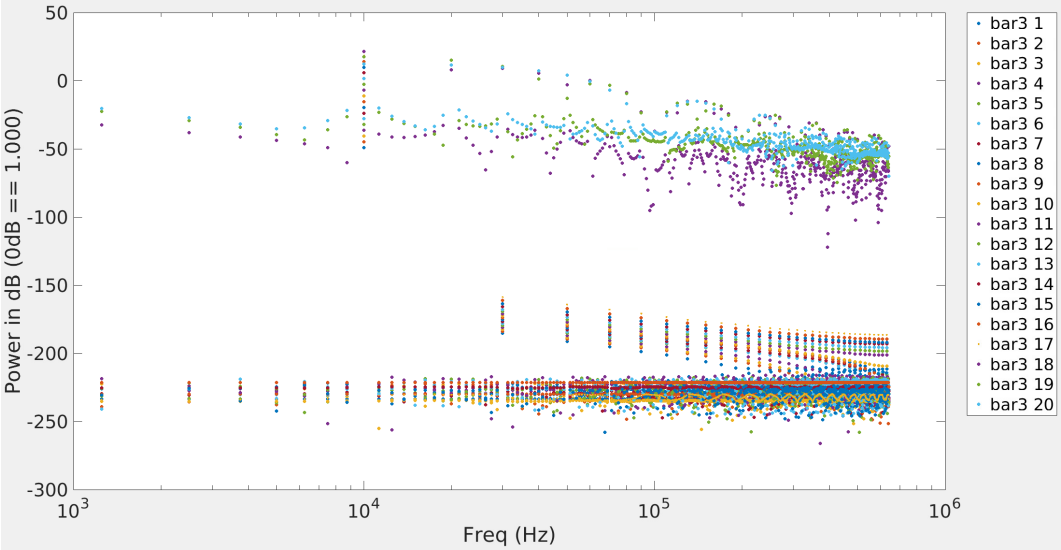


Figure 51: Frequency spectrum from sweeping from 1 mV to 10 V

The noise floor of the three highest input voltages is considerably higher than the rest at around 40 dB. This is because the output range of the op-amp in the amplifier is ±13.8 V, which means clipping will occur when the output voltage is supposed to go outside that range. This clipping appears as noise. Our feedback consists of an increase of five times the input due to the choice of resistance in the circuit meaning that 2.76V is our cutoff voltage for input voltage. The significant noise increase shown in Fig 51 shows up as a drastic decrease in SNR as seen in Fig 52.

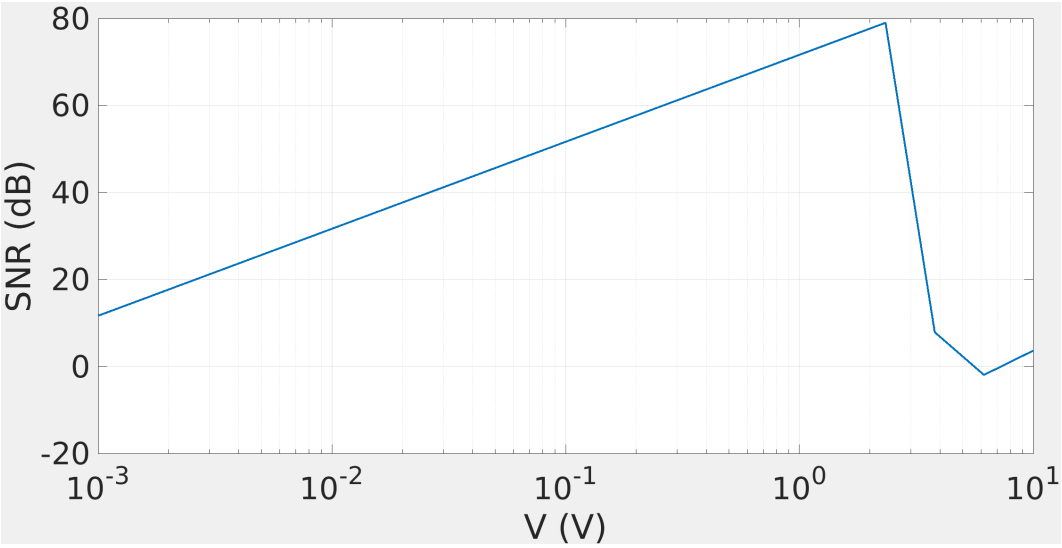


Figure 52: SNR from sweeping from 1 mV to 10 V

The reason why the SNR is so low at a low input level is because the signal is very close to the noise floor meaning that the more the signal strength increases the further away it comes from the noise floor. As mentioned before when clipping occurs as a result, the waveform gets distorted, leading to the creation of harmonics and additional frequencies not present in the original signal. Therefore when calculating the SNR the desired signal becomes masked by the distortion and noise.

5.3 Soft nonlinearity

Now in the circuit, a nonlinearity is introduced to the opamp whose behavior is unknown in this case. To inspect the effect of the opamp's nonlinearity which has been introduced, several sweeps were conducted on the opamp to inspect the new behavior of the circuit.

As for the first sweep, The voltage range of logarithmic steps from 1mV to 2.5V. The upper range of the sweep has been reduced to be able to see the effects on the harmonics without clipping. Except this nothing was changed except that the nonlinearity was introduced, simulating this produced the Fig 53 with the harmonics seen in 54.

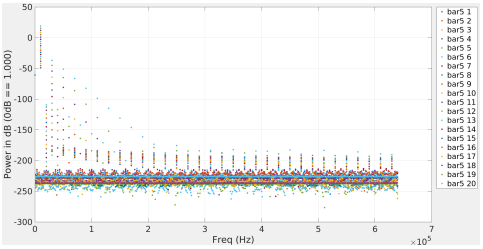


Figure 53: Spectrum with nonlinearities introduced

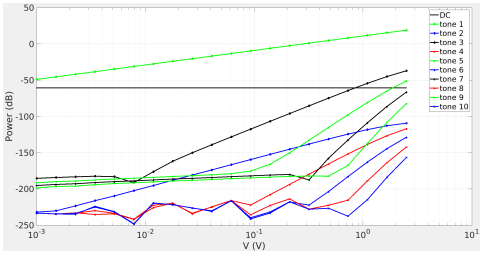


Figure 54: Harmonics of the schematic with nonlinearities introduced

As one can see from the graphs the nonlinearity seems to grow with the Voltage increase. Looking at the harmonics we can see that the dominating harmonic for this circuit seems to be Tone 3 (K=3) which is the first harmonic. As for the higher-order harmonics, one can see that the tones arise faster with increased voltage non-linearly. This further increases the theory that the nonlinearity introduced is increasing for larger voltage.

Also one can observe that the DC level in the plot Fig 54 that the DC level is the same for every input voltage. This is because the only DC component we have is coming from the Op-amp and isn't adjusted in this case.

For the next sweep that was conducted the AOLDC component in the Op amp was reduced to 20 dB. The same sweep was conducted with the same parameters as before creating the waveform charts Fig 55 AND 56.

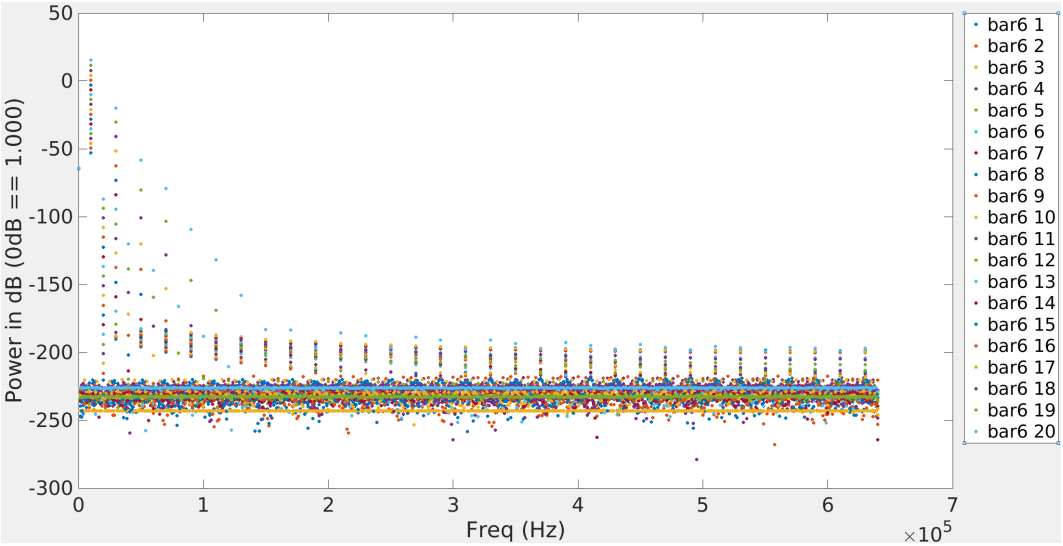


Figure 55: 20 dB gain spectrum with nonlinearities

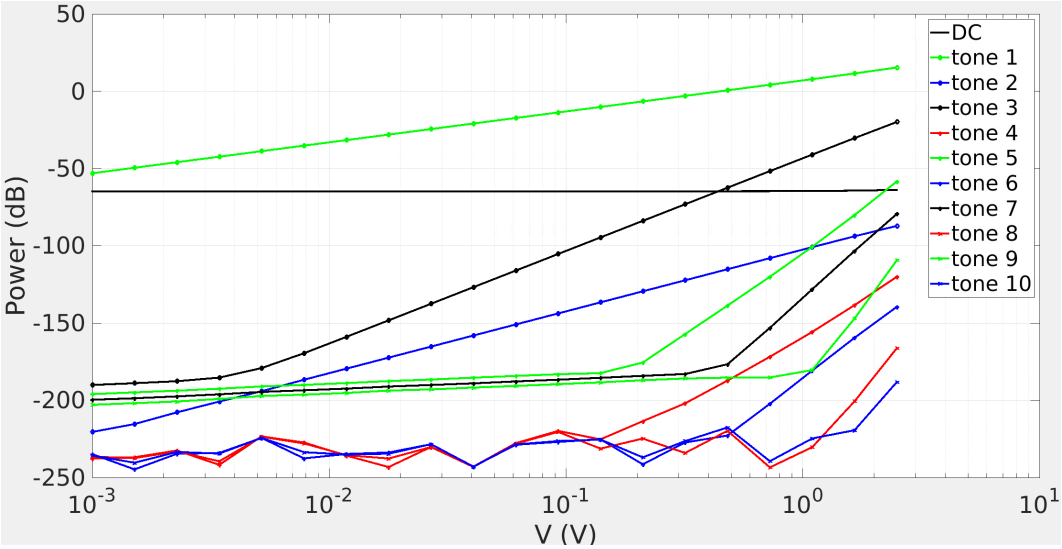


Figure 56: 20 dB gain harmonics with nonlinearities

From the resulting SNR figure, it becomes evident that clipping now occurs earlier in the circuit, attributed to a reduction in the op-amp's maximum gain. This acceleration in clipping causes the first harmonic to escalate more rapidly, resulting in a degraded (SNR) for the entire circuit. Notably, the SNR waveform has shifted to the left, reflecting the occurrence of clipping at lower voltages compared to the previous configuration.

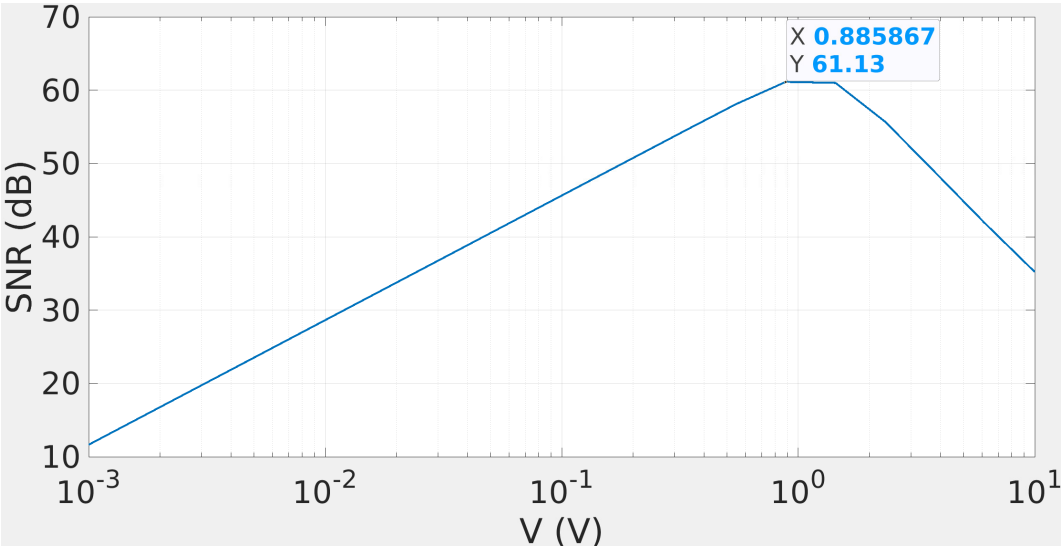


Figure 57: 20 dB Signal-to-noise ratio with nonlinearities

5.4 Conclusion

In conclusion, this report delved into the intricate aspects of circuit analysis, employing Cadence for data extraction and subsequent processing in Matlab. Further the examination of an op-amp circuit was analyzed, thus highligting the interplay between input voltage, clipping, and SNR degradation

The investigation into nonlinearities introduced an additional layer of complexity to the circuit's behavior. Thus examining an unknown characteristic in a circuit and the effect of this uncertainty, conducting different tests to see how the circuit would react.

6 Lab 6

The use of different sampling methods plays a crucial role in signal processing. This lab will be an insight into how different sampling methods can be used in combination with other techniques to reduce overall noise by eg introducing oversampling.

6.1 Oversampling

To test the effects of oversampling, the model shown in Fig 58 was created. It uses a 1-D lookup table for the A/D converter with 2^7 bits resolution making for a sampling frequency f_s of 128 Hz ($2^7 = 128$) and with an input of a 2 Hz sine wave. The continuous response and the frequency spectrum of the model are shown in Fig **??**.

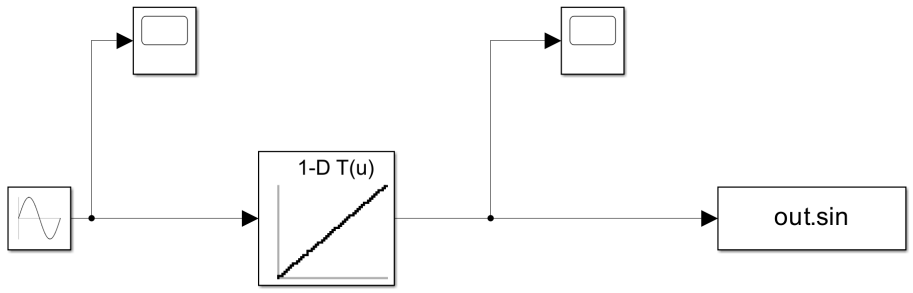


Figure 58: Test model for oversampling

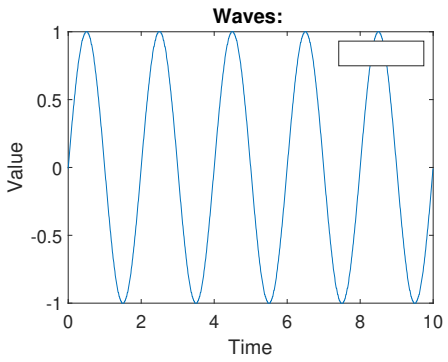


Figure 59: Waveform chart of the circuit

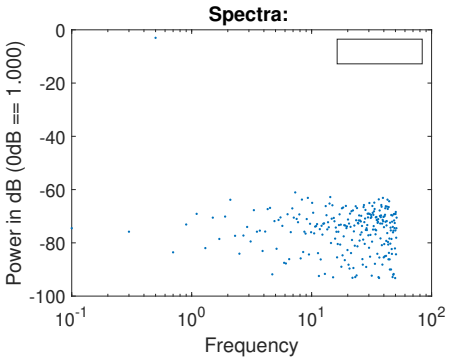


Figure 60: Frequency spectrum with 1024 data points

Now increasing the sample ratio of the model with a factor of 2^5 creates $2^5 = 32$ times more samples. Simulating this waveform and spectrum creates the figures in Fig 61

Now if we compare the over-sampled case with that of a normal sampled model we attain the same SNR. The SNR can be significantly improved however when considering that the bandwidth of the signal is constant while f_s becomes larger. A filter after the sampling, preferably in the digital domain to reduce cost, can remove the now many more frequencies which are outside the signal bandwidth. Since the SNR was constant despite there being frequency components at higher levels with the oversampling, the power of the noise at each component must be smaller. By including an, in this case, ideal filter the SNR was improved as seen in Table 5.

While an ideal filter could be placed directly outside the bandwidth of the possible input signals, it is not a very realistic scenario since a real filter would unavoidably need a frequency interval to have time to reach the desired attenuation. For simplicity's sake, this concern

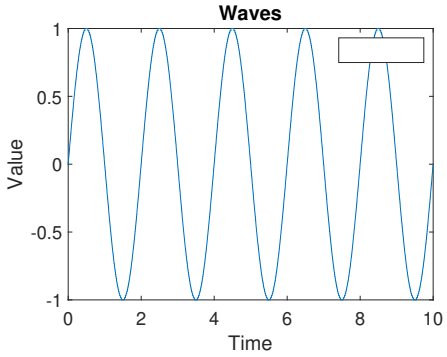


Figure 61: Spectrum with 1024 data points

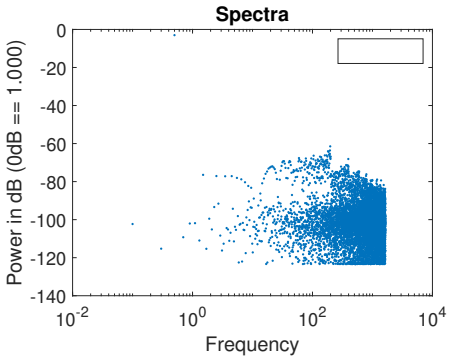


Figure 62: Waveform chart of the circuit

!h	No filter	Filter
	44 dB	59 dB

Table 5: SNR for an ideal filter for high frequencies when oversampling

is ignored however and frequencies above f_s of the non-oversampled model are filtered out meaning the only points taken into the calculation is the first 2^7 measurments. We should have made the filter begin at lower frequencies at half of f_s of the non-oversampled model since signals above the Nyquist frequency which is half f_s cannot be in the bandwidth assuming no undersampling and that the converter is working as intended.

In this case this seems to be the correct case since we can theoretically calculate the expected improvement as 15

$$20 \cdot log_{10}(\sqrt{2^n}) \tag{23}$$

$$20 \cdot log_{10}(sqrt(2^5)) = 20 \cdot 0.75 = 15dB \tag{24}$$

6.2 Single-pole noise shaping

In this section the SNR improvements from oversampling is improved by including noise shapening. To do this, the model shown Fig 58 was updated with a discrete integrator added in series before the quantizer and with a negative feedback loop to make the model in Fig 63.

By adjusting the input signal to be 1 dB lower than the power in section 6.1, the new amplitude of the sine generator becomes 0.89 unit power. Observing Fig 64 it's visible how the average of the two oscillating values is 0.88 meaning that the model fulfills the task.

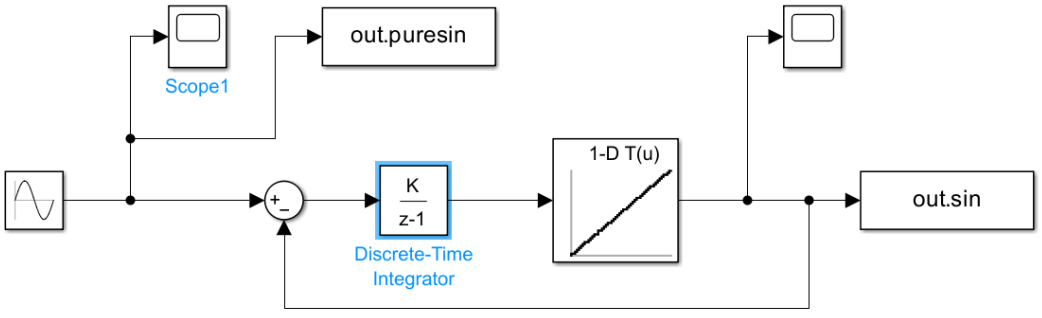


Figure 63: The model with an added discrete integrator

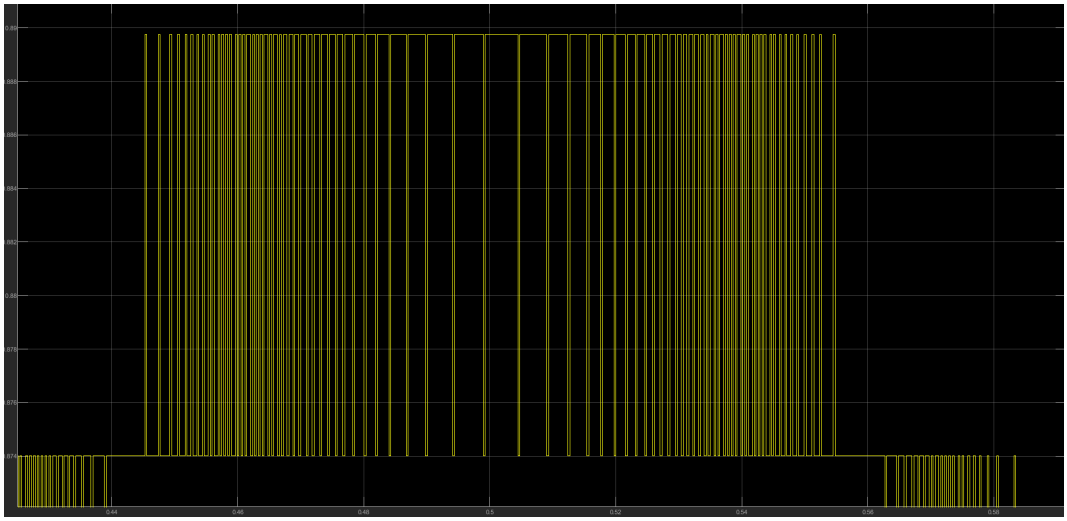


Figure 64: Waveform chart for a zoomed-in case for the model

No ideal filter	Ideal filter
41 dB	74 dB

Table 6: Oversampled signal with both non and ideal filtered SNR

Doing the same "oversampled" case as in the previous section Fig 66 was attained. Here it's observable how the integrator keeps adding up all of the samples making the noise stronger for higher frequencies.

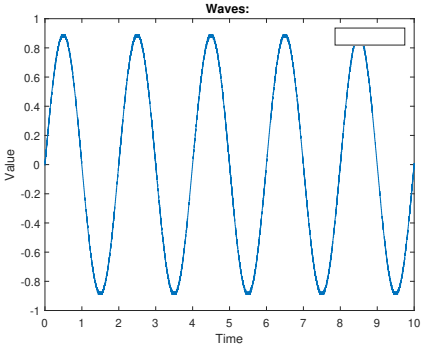


Figure 65: Spectrum with 1024 data points

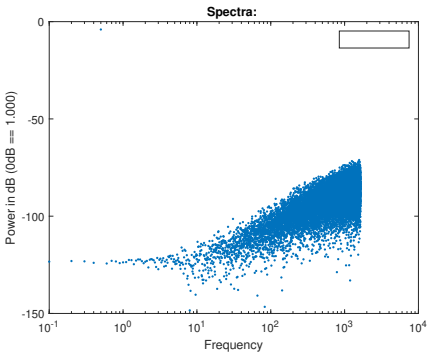


Figure 66: Waveform chart of the circuit

To same as in the previous chapter, we implement the "perfect filter" once again with this we calculate the SNR for both of these scenarios again and attain the Table 7. One can see here that for the case when all higher-order samples are cut the SNR increases substantially.

The observed increase in noise by 20 dB per decade, as depicted in Fig 66, aligns with the expected behavior in systems where noise contributions accumulate with frequency. In this context, the integrator in the system plays a crucial role by summing up noise contributions from various frequencies over time. Therefore, the observed trend of a 20 dB per decade increase in noise can be attributed to the integrator's effect.

6.3 Filters and oversampling

As seen in the previous section we saw an overshoot of noise at high frequencies a standard procedure is to implement a low-pass filter. This was done by implementing a Butterworth

filter of the third order. This is shown in Fig 67 where "Filter Designer" is the low-pass filter.

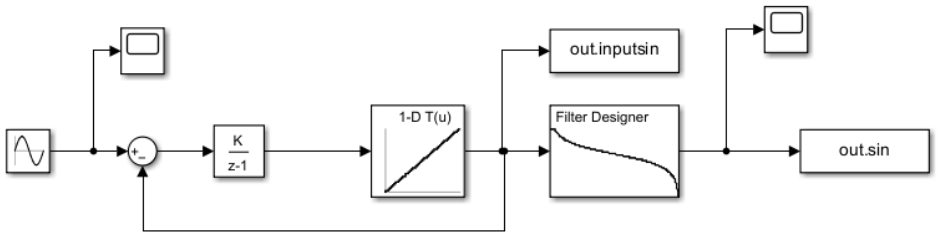


Figure 67: Circuit design with an added third order butterfilter

Further, we can observe in Fig 68 how the filter affects the system. From the spectrum its deductible to conclude that the Butterworth filter generates about a 60 dB drop per decade which is expected because it's a third-order filter. Comparing the SNR value with the "ideal filter" we get Table 7 which shows that the Butterworth filter is doing an almost ideal job at this point in removing the noise.

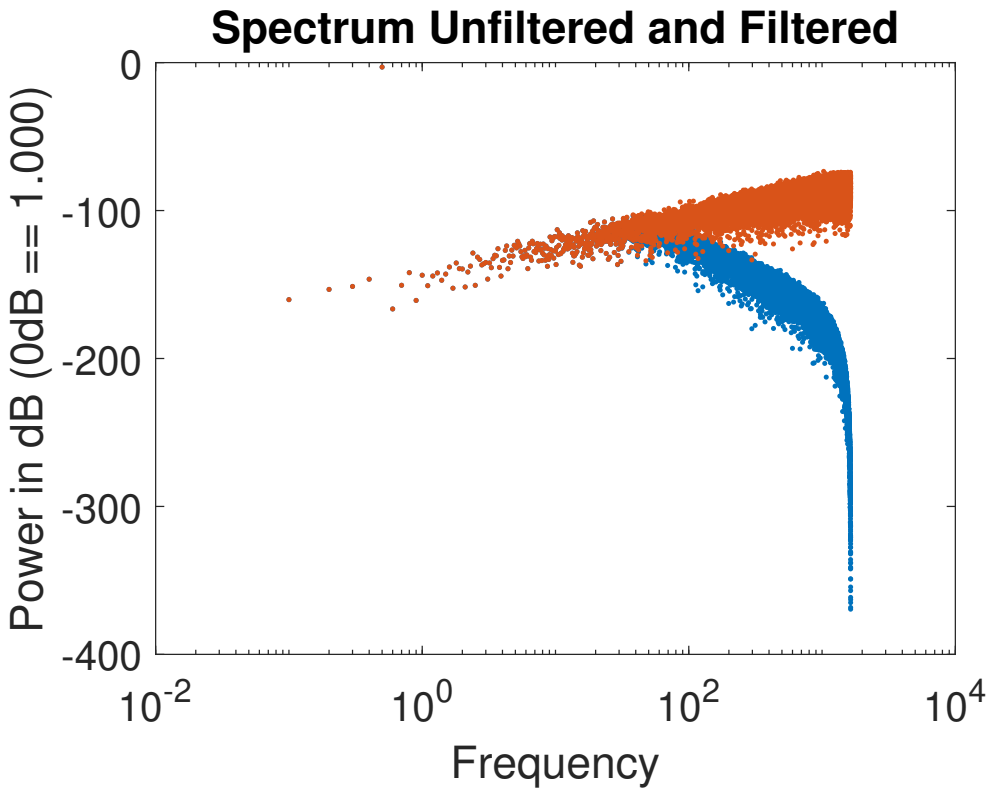


Figure 68: Filtered signal (blue) versus nonfiltered (orange)

When discussing the filter order we have to take into account two things. Firstly how much should the SNR be to acquire the desired result and secondly what is the maximum computational cost of the system? In this nonspecific scenario, it's observable from Fig 69 that the SNR flattens out between filter orders three and four which in this case should be an optimal choice. Since the cost exponentially increases with each order.

Filter	Ideal filter
81.1	99.9

Table 7: Table for a butterworth of third order and the ideal filter used previously

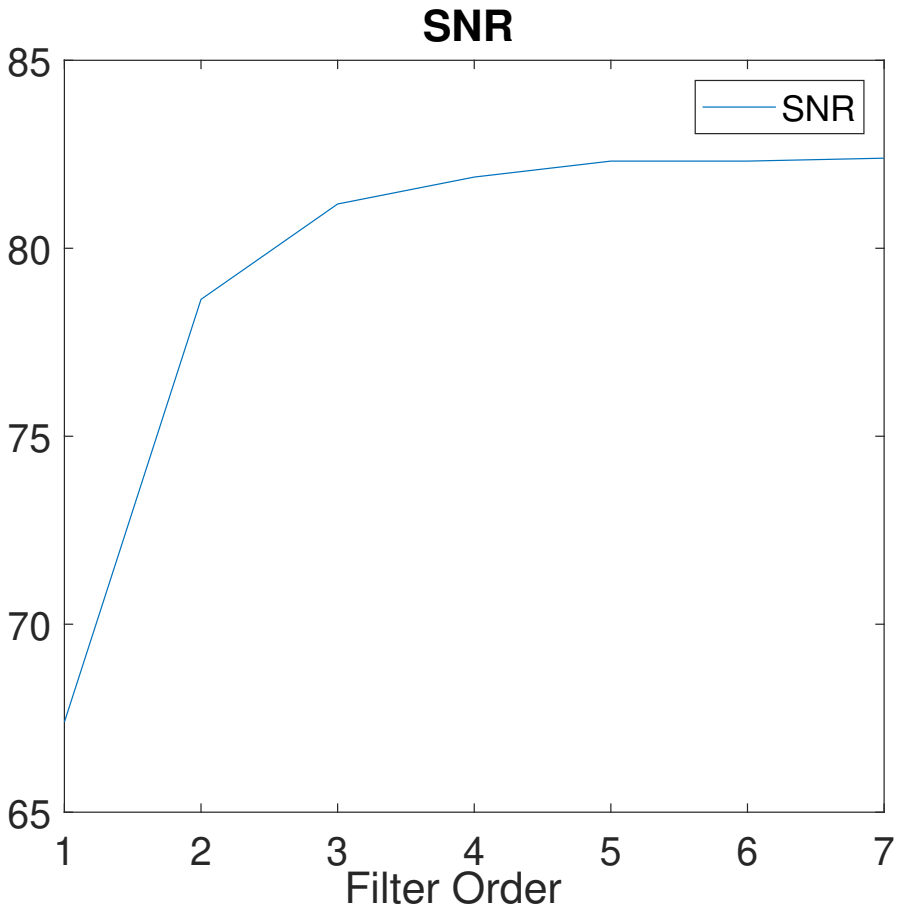


Figure 69: Depiction of the diminishing return when increasing order of the Butter worth filter

6.4 One-bit sigma-delta converter

For this last part, the system will only consist of a single digit quantizer meaning that it will only have two states. The resulting system will look very close to the previous one except for the lookup table shown in Fig 70.

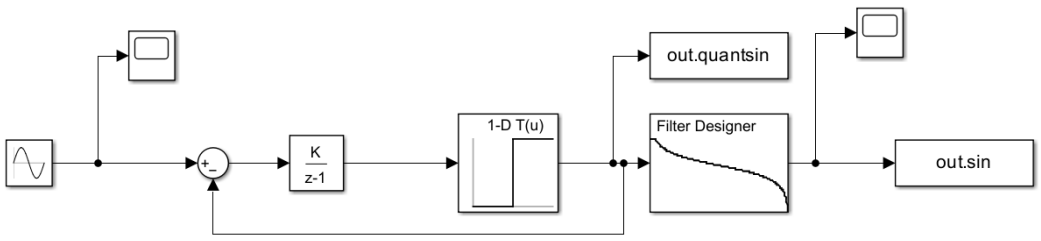


Figure 70: Circuit for one bit converter

Recalculating the SNR value with this circuit an SNR of 36 dB was attained. Furthermore, the next step will be to set the input amplitude to zero and as observable the signal

is not only gone but has been replaced by a single harmonic. This is shown in Fig 71 as seen this signal is not zero but it's very close to zero. This might be explained by the fact that there is some quantization noise which becomes more spread out. Thus this might be a single harmonic or a low-level residual signal.

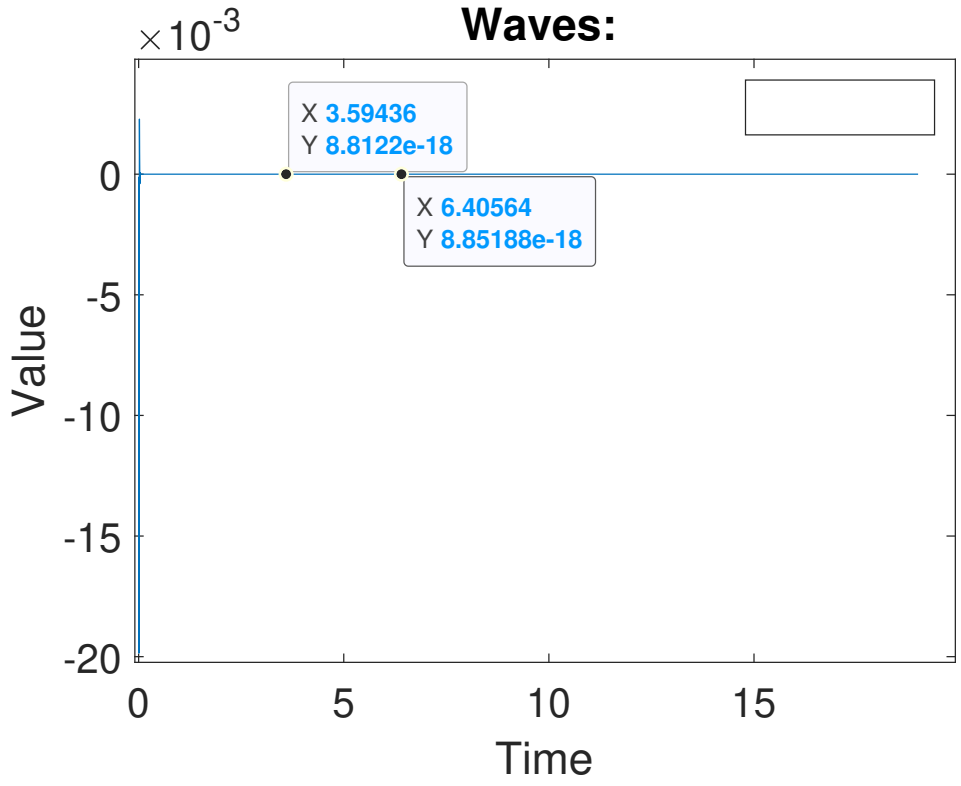


Figure 71: Waveform graph for sinus wave with 0 amplitude

Applying a bias with the DC component of 0.01 creates a new behavior in the system. As the filtered signal will only take values from -1 to 1 we the nature of this behavior with a frequency of the bits at around 3.28 KHz. this is explained by the fact that we have a sampling ratio set to 2^12 which corresponds to stepsize. This difference can be seen in the Fig 72

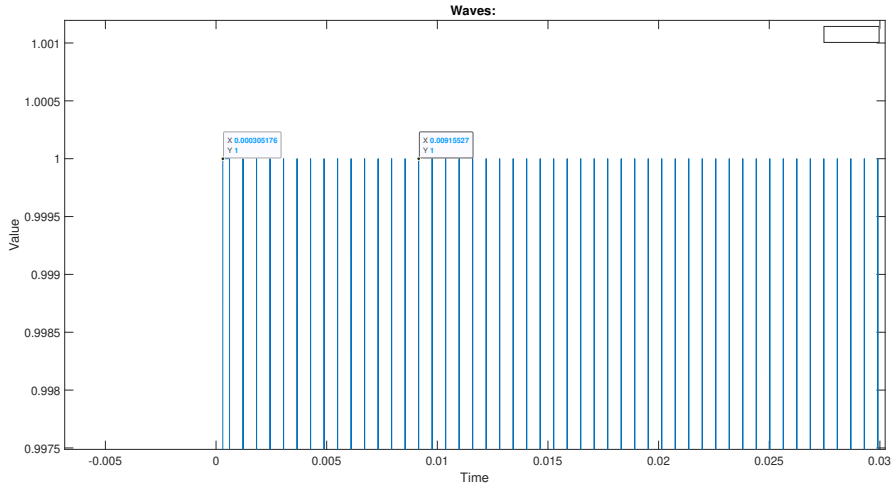
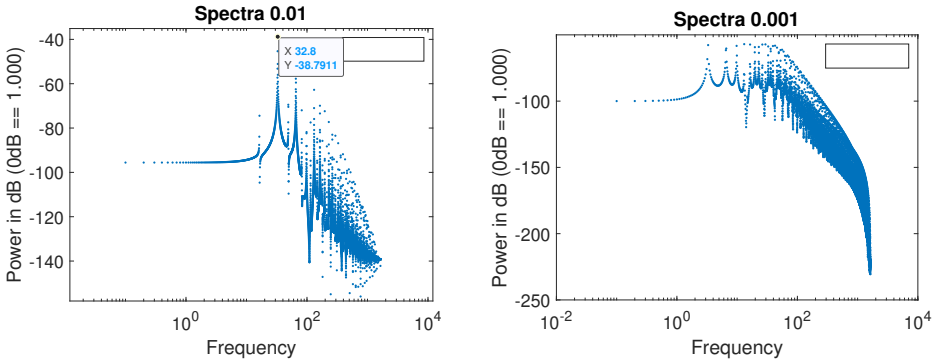


Figure 72: Waveform graph for the filtered sinus wave with an added DC component of 0.01 amplitude



(a) Spectrum for one step quantizer with DC 0.01 bias

(b) Spectrum for one step quantizer with DC 0.001 bias

Figure 73: Uniform quantization