Lab Project

Due Date: June 20, 2021

NATIONAL UNIVERSITY OF SCIENCES & TECHNOLOGY

Electrical Network Analysis (EE - 211)

Analog Anti-Aliasing Filter

Theory, Implementation and Results

Group

Muhammad Ahmed Mohsin	Muhammad Umer	Saad Bakhtiar	Tariq Umar
SEECS	*SEECS*	*SEECS*	*SEECS*
National University of Sciences and Technology	*National University of Sciences and Technology*	*National University of Sciences and Technology*	*National University of Sciences and Technology*
CMS: 333060	CMS: 345834	CMS: 341150	CMS: 334943

ABSTRACT

Computers process with Analog Signals by converting them to a Digital Signal first; They sample the original signal and quite often, there will be aliasing persisting in the sampled signal. Such aliasing gives rise to distortions in the desired output when it is converted back to Analog form. Such distortions can be removed by using a Low-Pass Filter with appropriate parameters. The effects of the filter itself were observed and dealt with the use of MATLAB and Simulink by visualizing and implementing a physical network. Moreover, a script was written to both simulate the original model and to plot the voltages that portray the effects of the filter on the output signal.

I. Abstract

In our Project we introduced the Anti-Aliasing filter and its simulation and code in MATLAB and SIMULINK. The Filter consisted of three voltage sources among which two were AC Sourced and one was a DC Source. Moreover, our circuit consisted of two second order op-amp non-inverting Sallen key filters. It is a two-stage network in which our first output is obtained without passing it through Anti-Aliasing filter and the second output was obtained after passing it through Anti-Aliasing filter.

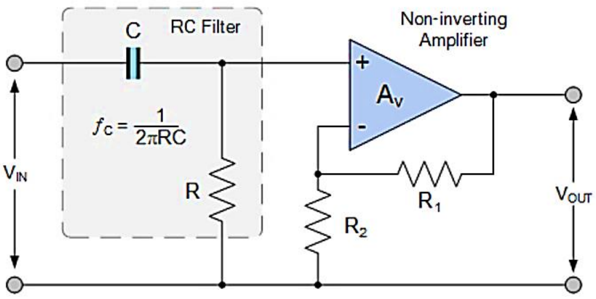
Moreover, our circuit consisted of Analog to digital converters. Analog-to-Digital converters (ADC) translate Analog signals, real world signals like temperature, pressure, voltage, current, distance, or light intensity, into a digital representation of that signal. It was used in order to display the output on the screen in form of digital screen.

Op-Amp amplifiers were used in two stages. Their gain was 1000. In order to have such a large gain we used it in two stages as a Sallen key filter. It is a low pass filter. The Sallen and Key topology is an active filter design based around a single non-inverting operational amplifier and two resistors, thus creating a voltage-controlled voltage-source (VCVS) design with filter characteristics of, high input impedance, low output impedance and good stability, and as such allows individual Sallen-key filter sections to be cascaded together to produce much higher order filters.

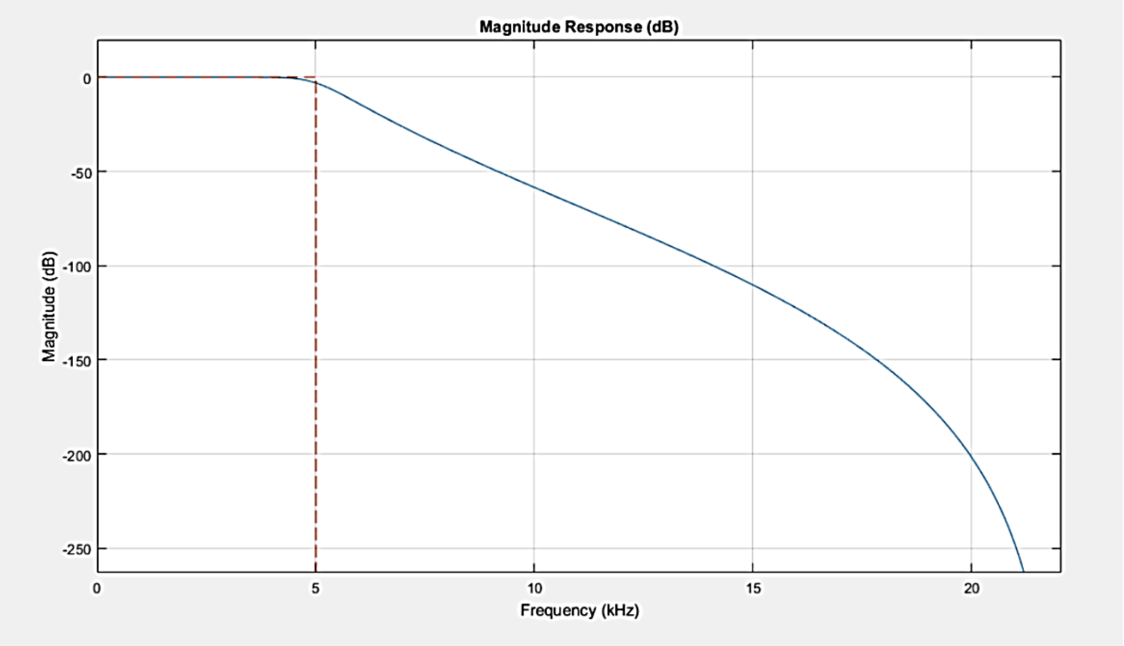
Sallen-Key is one of the most common filter configurations for designing first-order (1^{st}-order) and second order (2^{nd}-order) filters and as such is used as the basic building blocks for creating much higher order filters.

The main advantages of the Sallen-key filter design are:

- *Simplicity and Understanding of their Basic Design*
- *The use of a Non-inverting Amplifier to Increase Voltage Gain*
- *First and Second-order Filter Designs can be Easily Cascaded Together*
- *Low-pass and High-pass stages can be Cascaded Together*



As it is a low pass filter that we have used in our project the simulation for a general low pass filter on MATLAB is given as:



It allows the low frequency signals to pass through it but bocks the high frequency signals. That is why the graph decreases with increasing frequency. The simulation shown is of an ideal low pass general filter. The magnitude of the response decreases with the increasing frequency and after cut-off frequency it should ideally drop to zero but in real life the signal keeps gradually decreases to zero.

Next, we use the scope and the spectrum analyser to see our outputs as our graph. The scope is used to plot graphs of our input and outputs. A spectrum analyser measures the magnitude of an input signal versus frequency within the full frequency range of the instrument. The primary use is to measure the power of the spectrum of known and unknown signals.

While an oscilloscope displays a signal with respect to time, a spectrum analyser shows it with respect to frequency. Both of these tools are very important in any signal analysis application. Oscilloscopes are often used to get detailed timing information of a signal, or the timing relationships between several signals. We might use an oscilloscope to find the relative time delay between two signals. On the other hand, to observe the frequency properties of a signal, a spectrum analyser is required.

Procedure

First, we designed our complete circuit on Simulink and then we scripted our code on MATLAB. We used Simulink inbuilt blocks to build our circuit and matched its properties according to the requirement using the help of internet. Next, we implemented our code on MATLAB and defined our variables in the code which were used in Simulink.

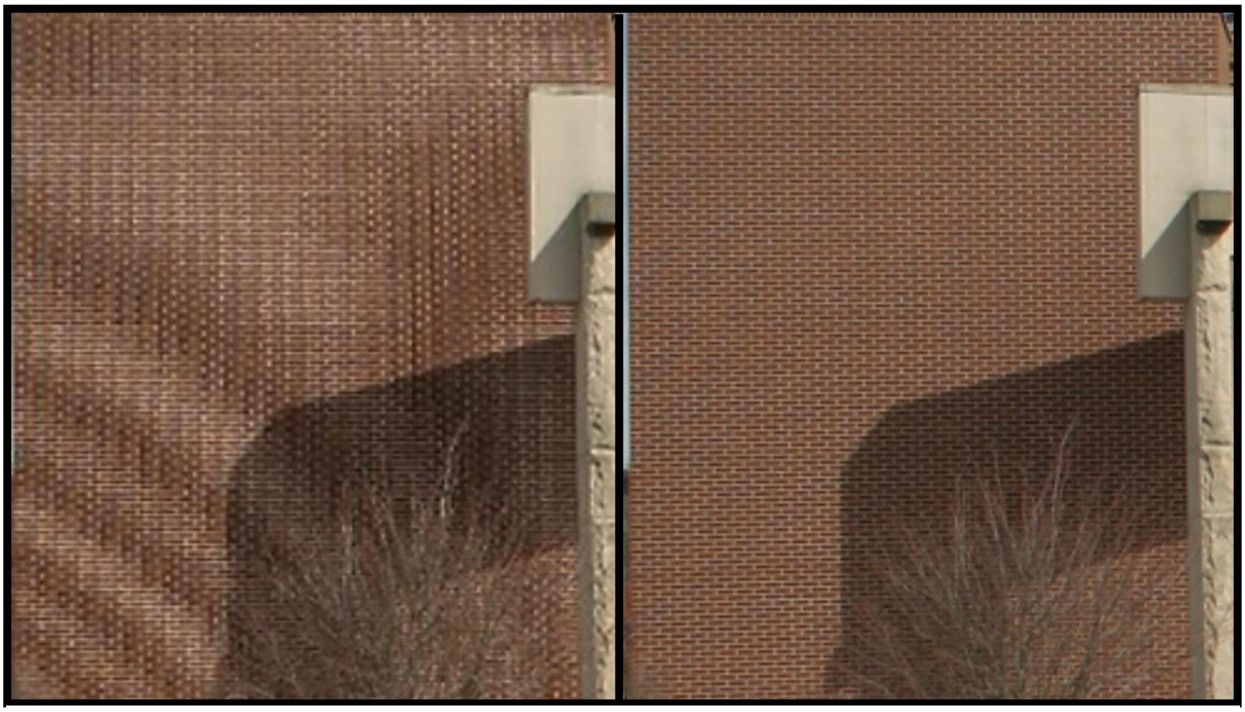
The values were random, so we took help form the internet. Next, we defined some commands in MATLAB code in order to plot the graphs. The script on MATLAB was linked with Simulink profile in order to save the variable names. Next, we observed graphs with different readings and the results are given in the next sections of the report.

Report Analysis

Our report Covers the synopsis of the project including the introduction and the procedure of the working. Next Section includes the Aliasing and the Anti-Aliasing filters defined. The next portion includes the Analog to digital converters explained. After explaining all the prerequisites, we explain the Analog anti-aliasing filter. It is the main portion of our report. This topic covers its built, its working, its function and Sallen key topology. After this we explain the results and conclusion and, in the end, we explain its applications.

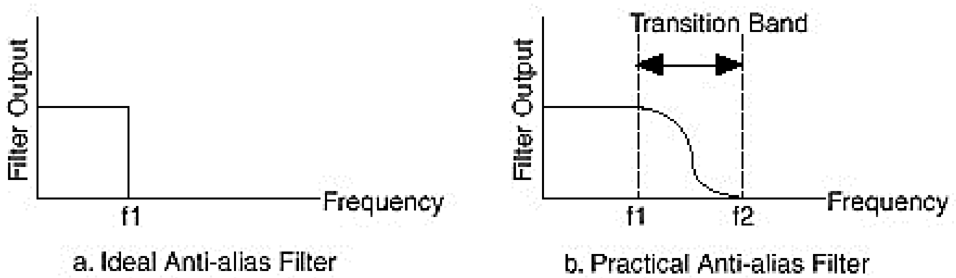
II. Anti-Aliasing

In signal processing, aliasing is an effect that causes different signals to become indistinguishable, when they are sampled (*a continuous-time signal is converted into a discrete-time signal*). It is also referred to the distortion that results when a signal, reconstructed from its samples is different from the original signal. Aliasing can occur in digital audio known as ***temporal aliasing*** and in digital images known as ***spatial aliasing***.



Without Optical Low-Pass Filter **With Optical Low-Pass Filter**

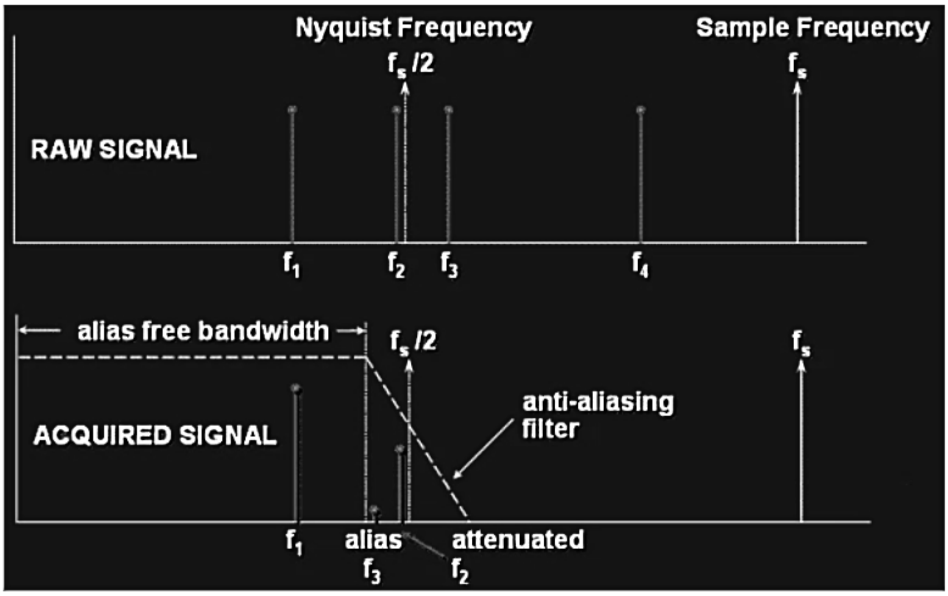
Aliasing can be avoided by using a low-pass or anti-aliasing (AAF) on the input signal before sampling to restrict its bandwidth to satisfy the Nyquist-Shannon sampling theorem for the required band. The anti-aliasing filter is an Analog filter.



The theorem states that unambiguous reconstruction of the signals from the samples is possible when the power of the frequencies above the Nyquist frequency is zero. The sampling rate should be twice the maximum frequency of the signal (or band) of interest.

A practical anti-aliasing filter permits some aliasing to occur, attenuate or sometimes distort the in-band frequencies close to the Nyquist limit. For this reason, many practical systems sample higher than that would be theoretically required by a perfect AAF to ensure that all frequencies of interest can be reconstructed, a practice called **oversampling**.

An illustration of an anti-aliasing filter being applied to a raw signal is shown below. Say that you want to sample *f1* and *f2* only. Note that *f3* lies in the transition band of the filter. Thus, the undesired frequency *f3* has been attenuated but its attenuated image still is sampled. Note also that *f4* has been eliminated because it lies above the transition band.



Example

An audio signal contains frequency component up to *20 KHz*. The Nyquist sampling theorem states a required sampling frequency of *40 KHz*. The anti-aliasing would have a cut-off frequency of *20 KHz*, but since this is not an ideal filter usually the sampling frequency used goes from *44.1 KHz* to *96 KHz*, allowing a transition band of at least *2 KHz*.

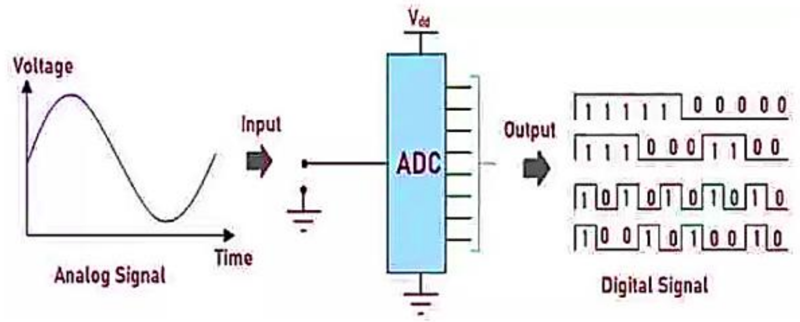
III. Analog to Digital Converters

As guessed by the name, the Analog to digital converter changes the Analog signals, such as a sound picked up by a microphone or light entering a digital camera, given in an input to digital signal. An ADC provides an isolated measurement such as an electronic device that converts an input Analog voltage or current to a digital number representing the magnitude of the voltage or current.

Working

In the real world, Analog signals are signals that have a continuous sequence with continuous values (there are some cases where it can be finite). These types of signals can come from sound, light, temperature and motion. Digital signals are represented by a sequence of discrete values where the signal is broken down into sequences that depend on the time series or sampling rate (more on this later). The easiest way to explain this it through a visual! Figure 1 shows a great example of what Analog and digital signals look like.

ADCs follow a sequence when converting Analog signals to digital. They first sample the signal, then quantify it to determine the resolution of the signal, and finally set binary values and send it to the system to read the digital signal. Two important aspects of the ADC are its sampling rate and resolution.



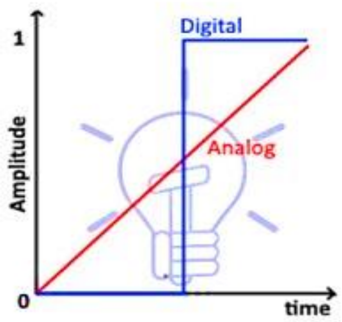
Work Flow of ADC

Analog to Digital Converter (ADC) basically converts physical variables which are Analog in nature to digital signal for processing. They have high conversion efficiency and requires less power. Examples of physical variables include audio signals, temperature, pressure etc.

Factors

Resolution of an ADC is the number of bits that represents the digital signal's amplitude. The Analog signal has continuous amplitude. It can have infinite values i.e. real, floating basically any value one can imagine. On the other hand, the digital signal has a discrete and finite number of values. These discrete values are represented using binary numbers (bits).

1-Bit Resolution



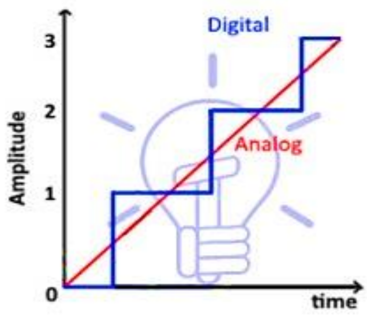
1-bit Resolution

The figure above shows an Analog signal represented in a digital form which is either 0 or 1. This is a 1-bit resolution. The resolution of ADC defines its number of steps.

Number of Steps = 2^n

Where n is the number of bits. Therefore, there are 2 steps in a 1-bit resolution.

2-Bit Resolution



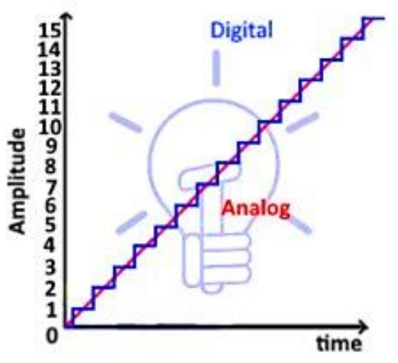
2-bit Resolution

This figure shows the conversion of Analog to digital in 2-bit resolution. There are 4 steps or quantization levels.

Number of Steps = 2^n = 4

Therefore, there are 4 steps in a 2-bit resolution.

4-Bit Resolution



4-bit Resolution

This figure shows 4-bit resolution. The number of steps in 4-bit resolution is 16.

Number of Steps = 2^n = 2^4 = 16

The number of steps increases exponentially with increase in the bit-resolution. It also implies that by increasing the bits of resolution the converted digital signal becomes more like the original Analog signal. So ideally, we can say that a digital signal with infinite resolution is an Analog signal.

Width of Step

The voltage difference between two adjacent steps is known as the width of the step. It is denoted by $\boldsymbol{\Delta v}$. So, a single step represents a fixed voltage that is

$$\boldsymbol{\Delta v = V_{ref}/2n}$$

Here, $\boldsymbol{V_{ref}}$ is the maximum voltage being converted & n represents the bits of resolution.

For example:

$\boldsymbol{V_{ref} = 10.24v}$ & $\boldsymbol{n = 10\ bits}$

Then:

$$\boldsymbol{\Delta v = 10.24/210} \qquad \boldsymbol{\Delta v = 10.24/1024} \qquad \boldsymbol{\Delta v = 0.01v}$$

Thus, the step-size or width of the step is 0.01v. In this ADC, a single bit increase represents a 0.01v of increase in the Analog input. If Analog input is increased by 0.01v then the output is increased by 1 bit.

Applications

In the modern world of growing technology, we are dependent on digital devices. These digital devices operate on the digital signal. But not every quantity is in digital form instead they are in Analog form. The applications of ADC are limitless. Some of these applications given below:

- Cell phones operate on the digital voice signal. Originally the voice is in Analog form, which is converted through ADC before feeding to the cell phone transmitter.

- Images and videos captured using camera is stored in any digital device, is also converted into digital form using ADC.
- Medical Imaging like x-ray & MRI also uses ADC to convert images into Digital form before modification. They are then modified for better understanding.
- Music from the cassette is also converted into the digital form such as CDs and thumb drives using ADC converters.
- Digital Oscilloscope also contains ADC for converting Analog signal into a digital signal for display purposes & different other features.
- Air conditioner contains temperature sensors for maintaining the room temperature. This temperature is converted into digital form using ADC so that onboard controller can read & adjust the cooling effect.

In today’s modern world almost, every device has become the digital version of itself & they need to have ADC in it. Because it has to operate in digital domain which can be only acquired using Analog to digital converter (ADC).

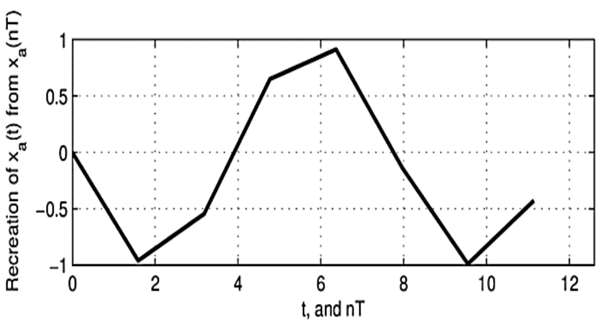
IV. Sampling and Aliasing

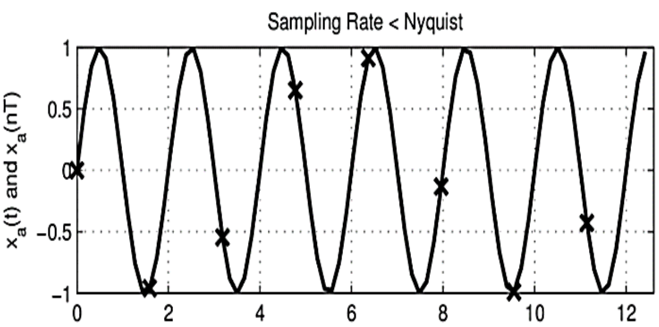
To understand sampling, we shall consider an example in which we suppose that you are recording a wave. Now to record a wave, you need to number points in a graph to get its actual shape. Suppose we are making a sin wave. Now to make a sin wave all we need is the frequency and amplitude of the wave. Now if we want to construct the original wave back, we need to provide digital signals to the data acquisition software which will use the data to produce the required output. In order to get the exact wave, output we need at least twice the sampling frequency and the original frequency.

$$F_S = 2\ F_{MAX}$$

Sampling Rate

If we provide more data than the than the original wave then it will construct the exact graph. If the data is less than Maximum frequency it will connect points in such an order which will not adhere to the shape of the original wave.





The wave form will appear of the frequency less than the original frequency. Thus, we need at least twice the sampling rate. However, increasing the sampling rate from more than twice will bring accuracy in our results. That appearing of the signal as allowed frequency signal than the original signal is called Aliasing.

Nyquist Frequency

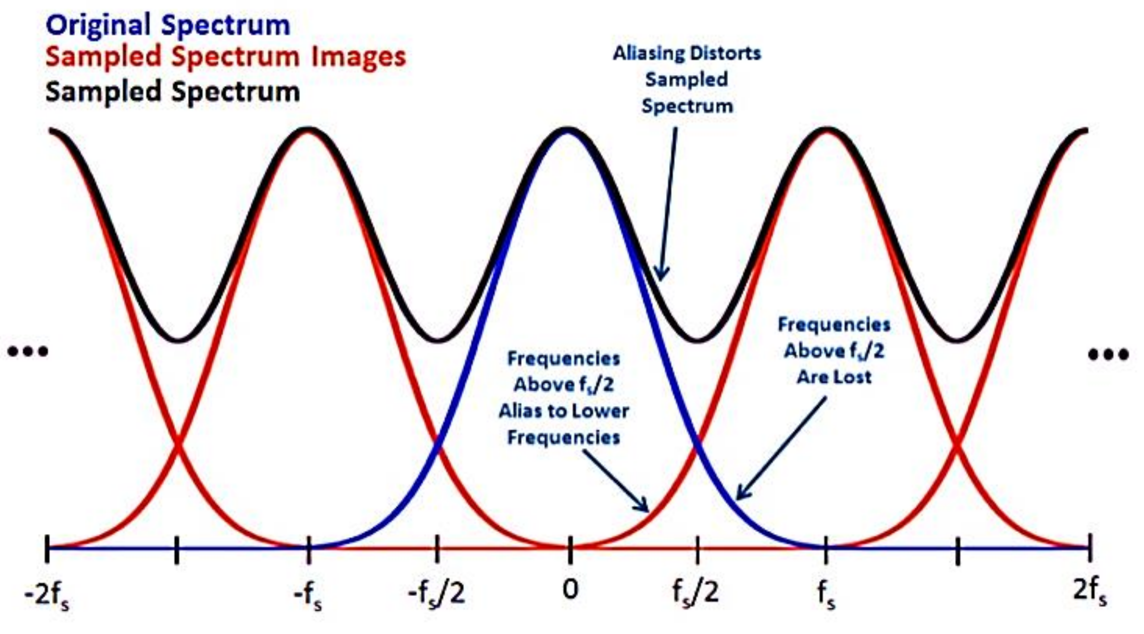
'It is the minimum rate at which a signal can be sampled without introducing errors, which is twice the highest frequency present in the signal.'

Now the question arises that how we prevent the signals greater than sampling rate to enter our data acquisition software. For that purpose, we use low pass filter which are basically designed to stop allow certain frequencies and stop the rest of them. If a signal greater than the sampling rate enters into our system thus, we will get wring readings.

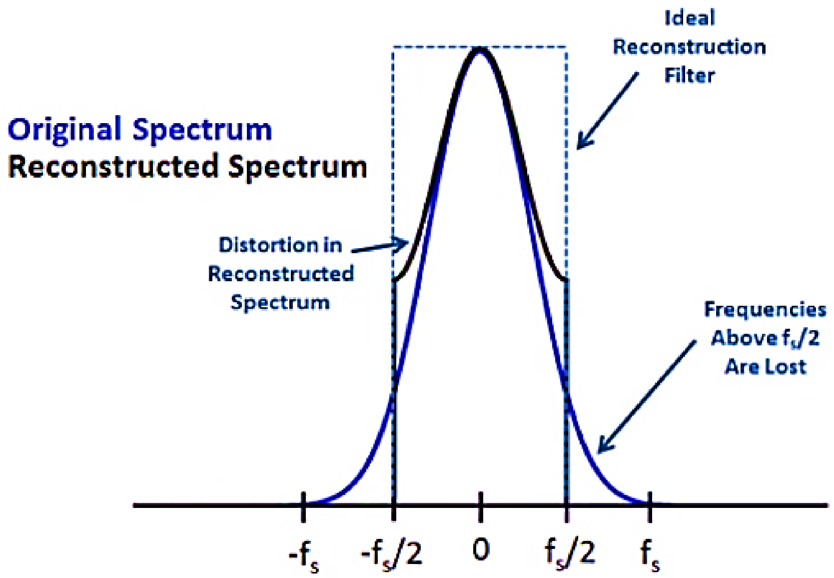
Now suppose that we are receiving an input signal. For example, music from our EarPods; We are receiving signal in time domain and suppose that we want to change signal to frequency domain. Now suppose that this signal has a frequency of Fs/2 which may cause aliasing. Although it contains information, but we need to stop that data from entering into our data acquisition software or it may cause aliasing.

By using a low pass filter, it will make the signal greater than Fs/2 go flat before it reaches the data acquisition software. If we use a first order filter circuit the graph will drop off at the rate of 20db/ decade. By using higher order filter circuits, it will drop off very fast.

Everything above the frequency of Fs/2 will fold over or what we call as Alias. Now consider the signal below:



The signal in the blue is the original signal in the frequency domain and we want to plot it in the time domain. The signals in the red show the sampled spectrum images in with a shift of Fs. The spectrum of this overall signal is the sum of all of these curves. Thus, there are infinite number of curves that are overlapping which will add up to give the one single point. These infinite number of frequencies when they add up, they mess up the original signal and what we call as aliasing.

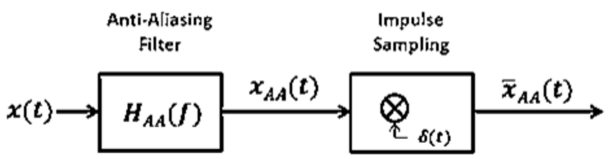


Now suppose that we signal our signal at Fs and then we tend to go back to our original signal? We put a reconstruction filter that eliminates everything above Fs/2 and everything below Fs/2. Well we are not getting the original graph at Fs/2 or -Fs/2 . What we are experiencing there is Aliasing. We don't get back to where we started but we get back that black curve instead.

Distortions

The distortion at **Fs/2** or **-Fs/2** is due to the folding of frequencies or Aliasing. All the frequencies have folded down and distorted down the things. There is a second type of distortion that our signal data greater than Fs/2 and less than Fs/2 is lost due to the reconstruction filter. Well, we can deal with the one of the distortions. We use the Anti -Aliasing filter to remove the folding signals.

V. Anti-Aliasing Filter



If we are going to sample at Fs then our total work lies between Fs/2 and -Fs/2.Anything above Fs/2 will fold down and distort our signal. The job of the Anti-Aliasing filter is to remove those frequencies. So, before I signal, I am going to get rid of all the frequencies above Fs/2 so that they will not fold down and cause aliasing.

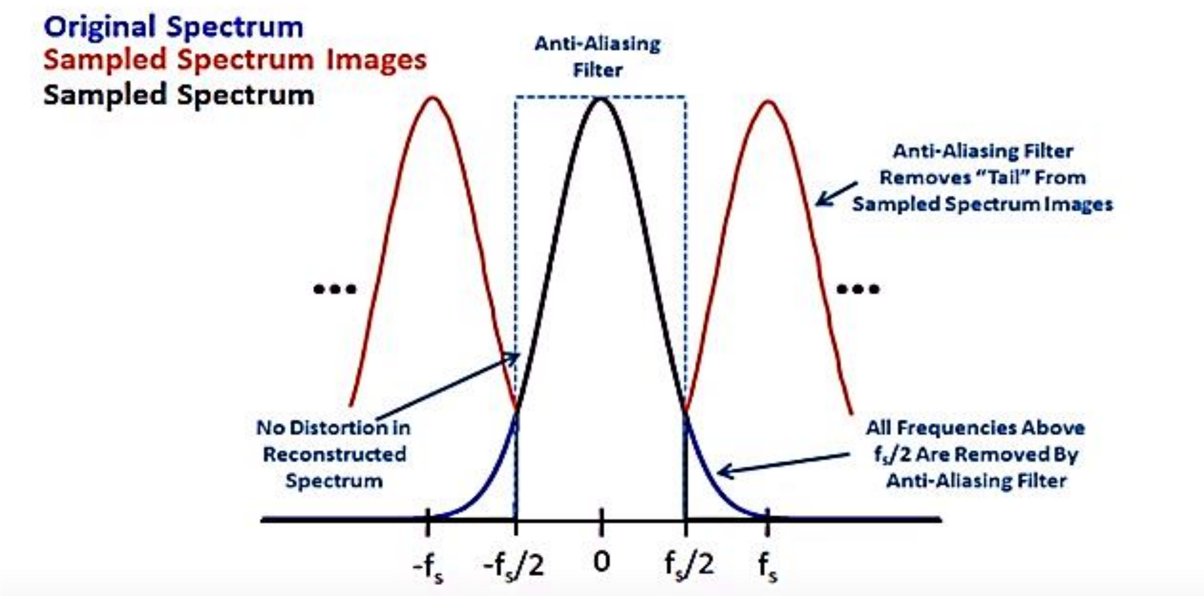
Another way of thinking it like we will limit the signal of what we call as the effective bandwidth. Although the input signals have a frequency content from infinity to minus infinity but after some time that content does not remain 'effective'.

Block Diagram

Once the signal passes over the Anti-Aliasing filter it chops of all those signals whose frequency is greater than Fs/2. Now this signal becomes an ideal low pass filtered signal. Now we pass it through the impulse sampling which is going to multiply our signal with the impulse string g(t). This will result in an impulse samples signal.

We need to understand that the fact the by introducing the Anti-Aliasing filter although I am removing one type of distortion, but I am also introducing distortion by removing signals greater than Fs/2 and less than -Fs/2. Thus, we need to be pretty accurate of what our effective bandwidth should be in order to remove least data of our signal and prevent anti-aliasing from happening.

After passing it though the Anti-Aliasing filter our distortion at Fs/2 and -Fs/2 is gone as shown below:



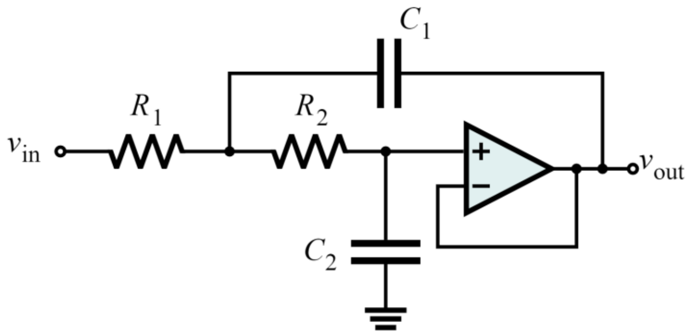
Our original blue spectrum and the black spectrum are now matching very well. Now I am going to transform the signal to my required frequency domain spectrum. Before sampling the signal in frequency domain spectrum, we passed it through the anti-aliasing filter in order to remove the distortions. Note that the above shown filter has an ideal low pass filter response.

VI. Sallen-Key Topology

The **Sallen-Key Filter** design is a second-order active filter topology which we can use as the basic building blocks for implementing higher order filter circuits, such as low-pass (LPF), high-pass (HPF) and band-pass (BPF) filter circuits. It is one of the most widely used filter topologies. The advantage of using Sallen-Key Filter designs is that they are simple to implement and understand.

Sallen-Key topology is an active filter design based around a single non-inverting operational amplifier and two resistors, thus creating a voltage-controlled voltage-source (VCVS) design with filter characteristics of, high input impedance, low output impedance and good stability, and as such allows individual Sallen-key filter sections to be cascaded together to produce much higher order filters.

Second-Order Low-Pass Filter by Sallen-Key Topology



Unity-Gain Low-Pass Sallen-Key Filter

This circuit represents a unity-gain low-pass configuration, having an operational amplifier as the buffer. The transfer function for this circuit will be:

$$\boldsymbol{H(s)} = \frac{\omega_o{}^2}{s^2 + 2\alpha s + \omega_o{}^2}$$

Here,

$$\omega_o = 2\pi\, f_o = \frac{1}{\sqrt{R_1 R_2 C_1 C_2}}$$

$$2\alpha = 2\zeta\omega_o = \frac{1}{C_1}\left(\frac{1}{R_1} + \frac{1}{R_2}\right) = \frac{1}{C_1}\left(\frac{R_1 + R_1}{R_1 R_1}\right)$$

We also know that:

$$Q = \frac{\omega_o}{2\alpha} = \frac{\sqrt{R_1 R_2 C_1 C_2}}{C_2(R_1 + R_2)}$$

This Q factor determines the height and width of the peak of the response of the filter. As this factor increases the resonance of the circuit will increase.

Working

1. **Low frequency Signals**

   For low frequencies, the capacitor will act as an open circuit. Thus, we will simply have R_1 and R_2 connected in series with the input impedance of amplifier, resulting in a buffer amplifier with gain of unity.

2. **High frequencies Signals**

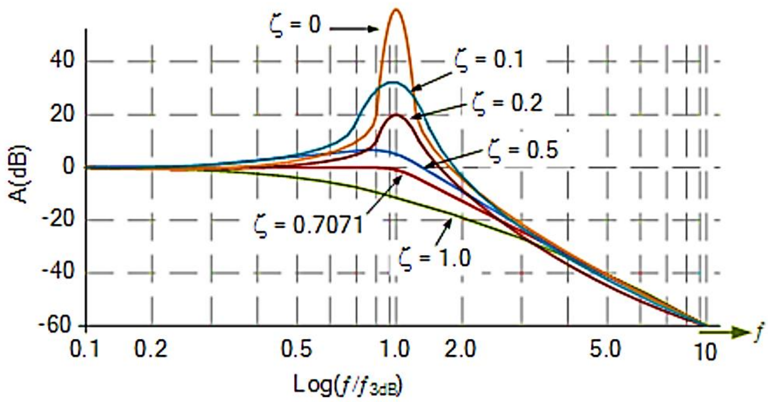
   For high frequencies, the capacitor will act as a short circuit. Now if C1 and C2 act has short circuit then the input to the amplifier will be 0, meaning the output will also be 0. Thus, we can see that the C_2 is connected to a virtual ground, which gives us a two pole RC filter and we will have a -40dB/decade slope.

3. Middle frequencies Signals

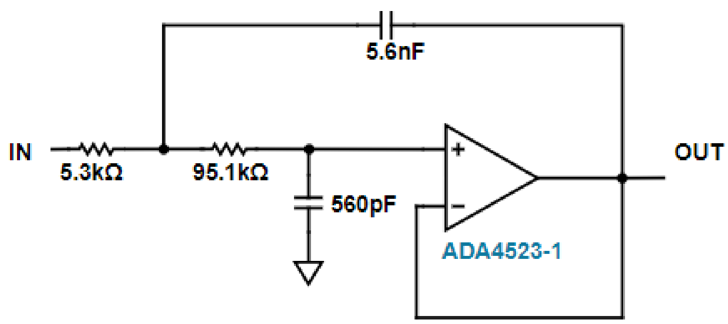
For middle frequencies, we will have some gain (peaking) due to the positive feedback from the output through C_1. Thus, the feedback will be dependent on the value of C_1. If we increase the value of C_1, its impedance will decrease that will increase the feedback. It can be confirmed by the formula of Q (peaking).

$$\boldsymbol{Q = \frac{\omega_o}{2\alpha} = \frac{\sqrt{R_1 R_2 C_1 C_2}}{C_2(R_1 + R_2)}}$$

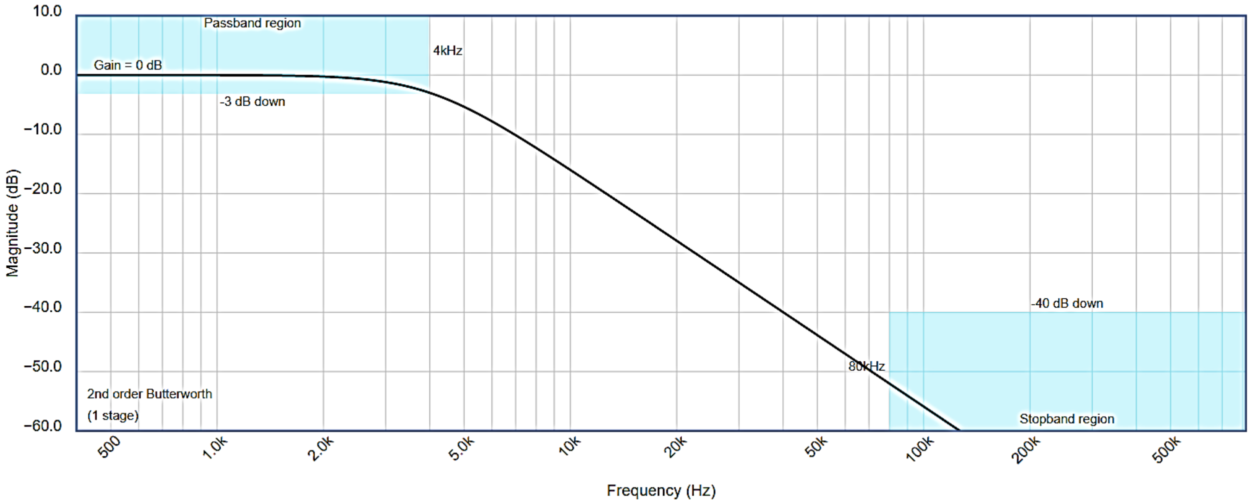
$$\mathbf{Q \propto C_1}$$



Example



Plot



This circuit have a cut off frequency of 4kHz and -50dB attenuation till 80kHz.

Higher Attenuation

We can increase the **attenuation rate** by using filters of higher order, which can be done by cascading more second order or first order filters.

Example

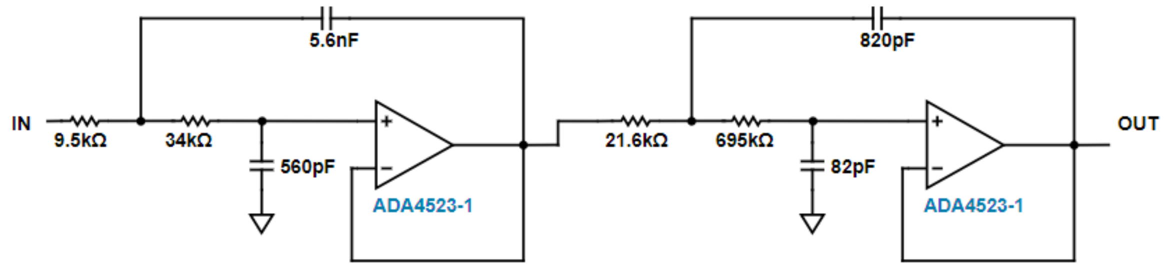
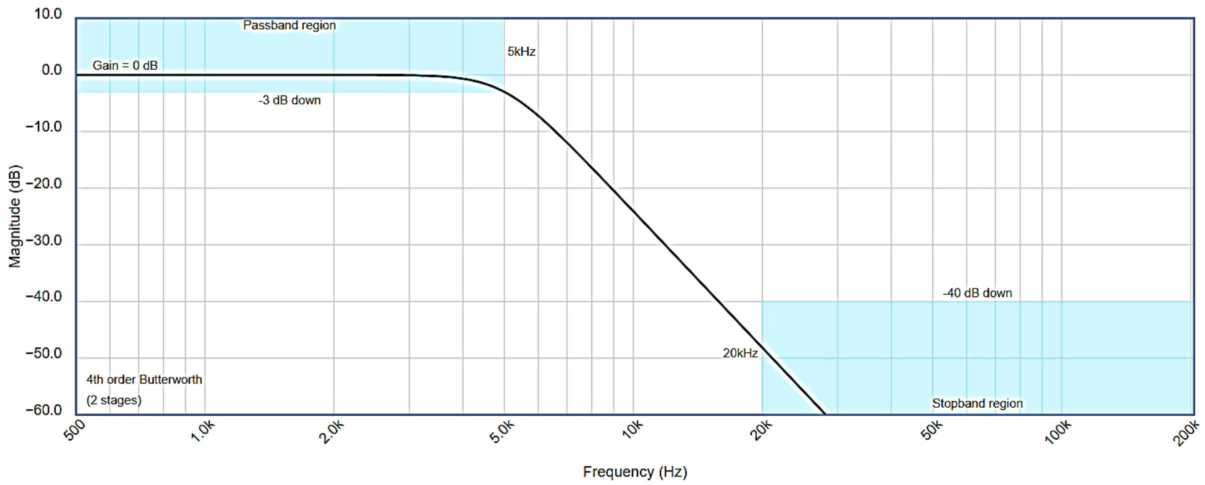


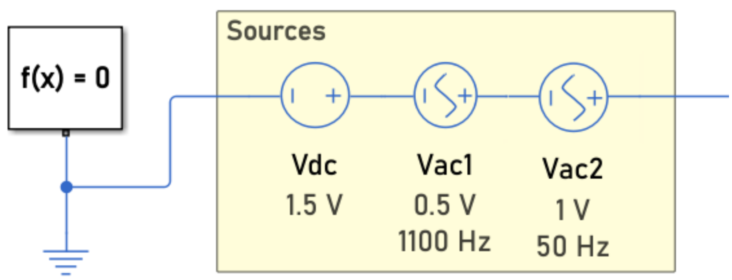
Figure 3: Low-Pass Sallen-Key Filter in Two Stages

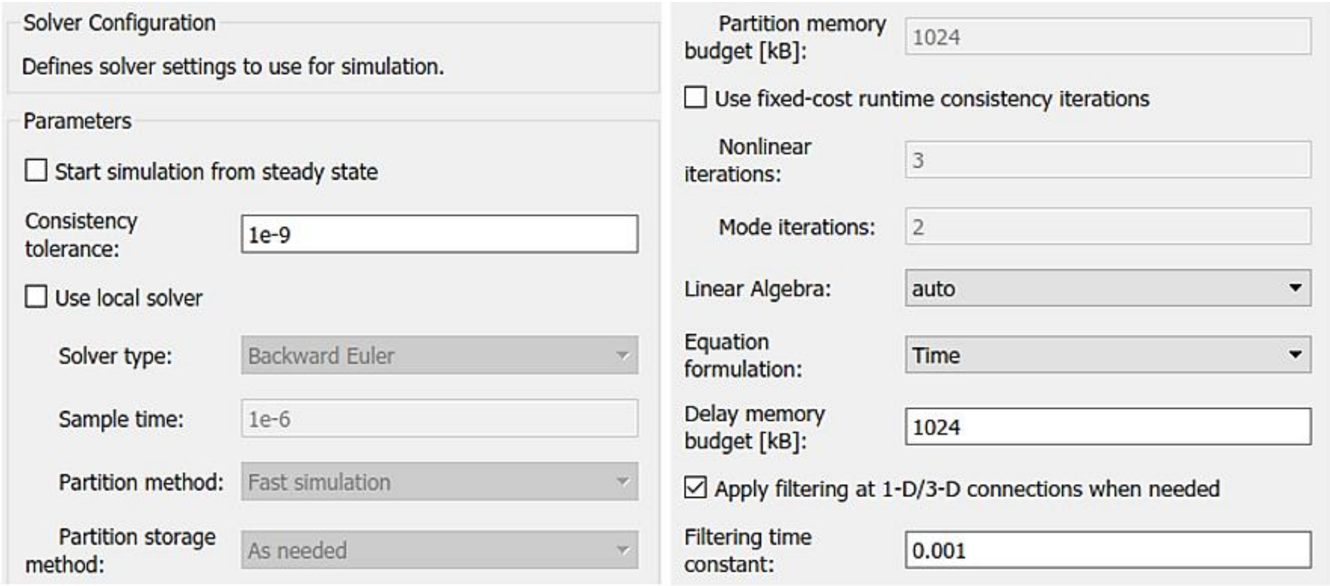


This circuit have a cut off frequency of 5kHz and -50dB attenuation till 20kHz.

VII. Simulink Model and Simulations

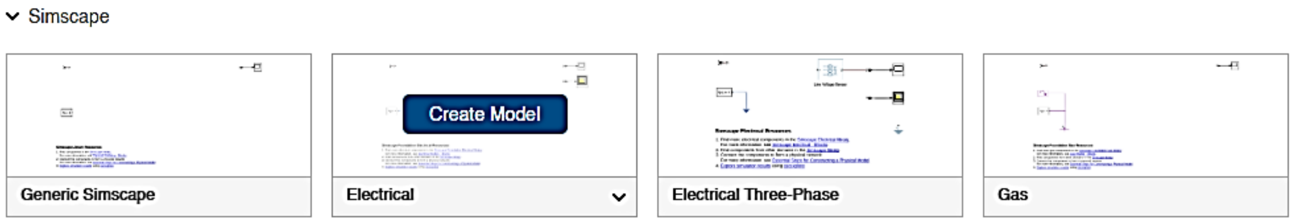
So, now that we’ve covered the theoretical aspects of our Analog Anti-Aliasing Filter, we can proceed to the implementation of our circuit on Simulink and create a supporting Script to plot the Voltages and to simulate the model itself. Following figure displays our input section of the complete model. It consists of three Voltage sources, a Solver Configuration and an Electrical Reference.





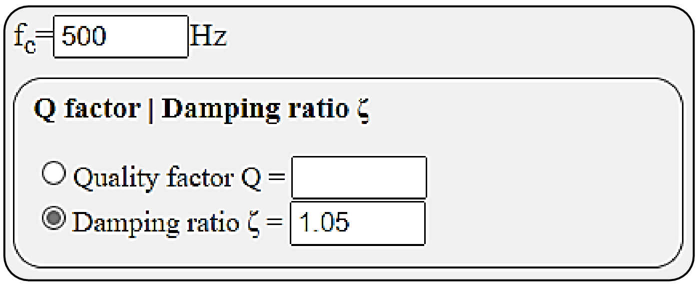
Solver Configuration

The Electrical Reference and the Solver Configuration along with its parameters are set to the given figure values when creating a model on the template of Electrical Simscape. This template can be accessed through the New → Template → Model



Which concludes implementation of an Analog input. Our desired signal is a 50 Hz AC Source along with a 1100 Hz AC Source that does not get captured by the 1 kHz Sampling Frequency of our ADC Converter.

Now, we move onto the model of our actual filter which is being used to remove aliasing and give us the correct Signal on the Scope. We also desire to remove as much of 'curving' as we can on our Output of Spectrum Analyzer. We find the values of our components either by Mathematics or using any Optimal Filter Designer Online.



Using the OKOWA Filter Designer Tool, we design a 2nd Order Low-Pass Filter with optimal values to correspond with a damping ratio of 1.05. Using this tool, we get the following Transfer Function and values of components in our LP Filter.

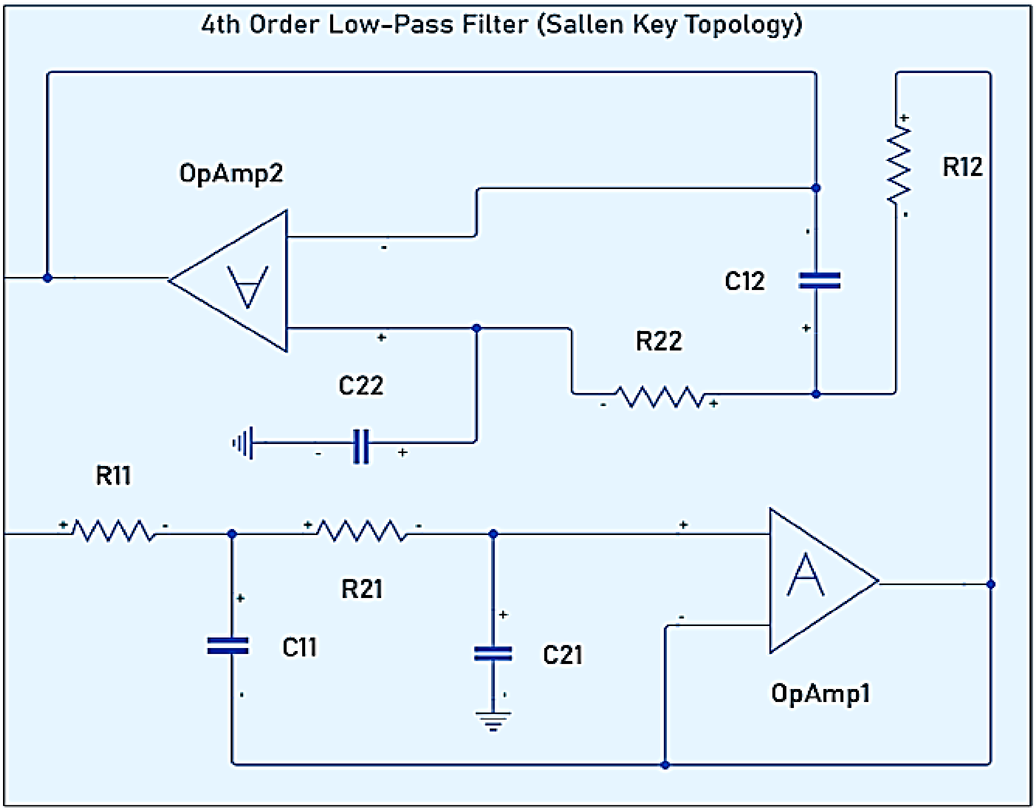
Transfer Function:

$$G(s)=\frac{9689922.4806202}{s^2+6492.2480620155s+9689922.4806202}$$

```
Components
R1 = 43 kΩ
R2 = 24 kΩ
C1 = 0.01 uF
C2 = 0.01 uF
```

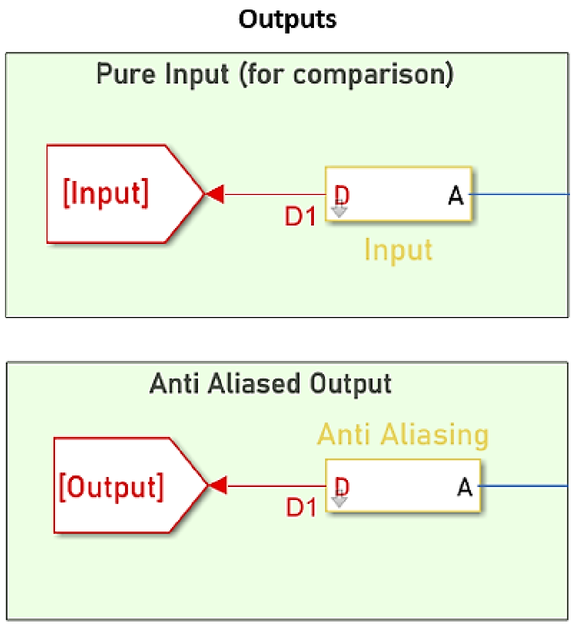
The latter part of the annotations of the Capacitors and Resistors is based upon the fact that our Filter is a cascaded form of two 2nd order Sallen-Key Filters. i.e.

```
R11 – Resistor 1 of the 1st Stage
C12 – Capacitor 1 of the 2nd Stage
```

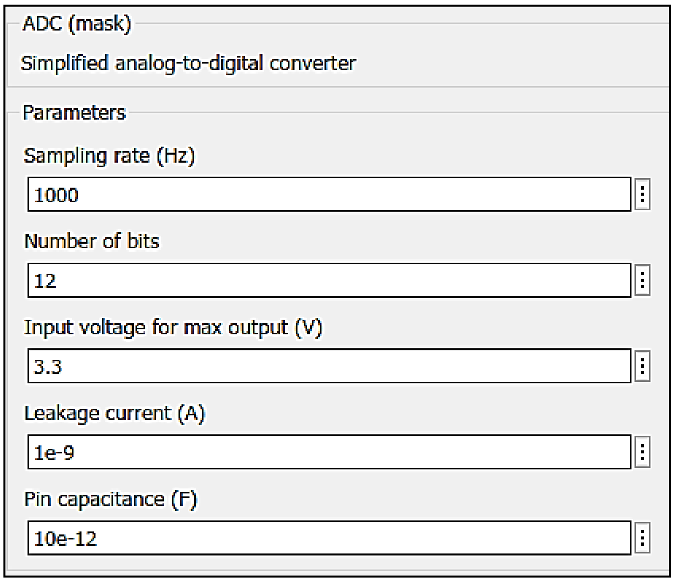


4th Order Low-Pass Filter (Sallen Key Topology)

This the schematic diagram of our Anti-Aliasing filter and is solely responsible for altering and reducing aliasing in the input Analog signal. It is at its core utilizing the one of the biggest advantages of Sallen-Key topology; that higher order filters can be easily made by cascading 2nd order filters and is the reason why our Frequency Spectra displays all the harmonics in accordance with reduced levels of distortions. It also contributes to the fact that a higher gain can be attained when two are cascaded together that cannot be attained through a single Op-Amp as it will render our system unstable.

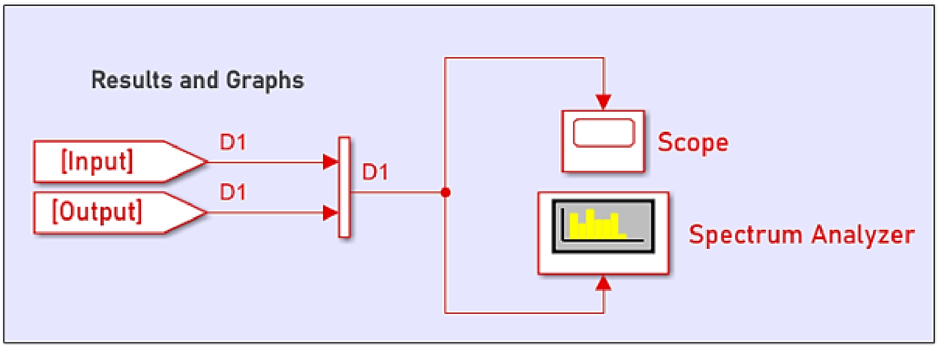


Note that the Signal that is being fed to the ADC Component is Analogous in nature as it was generated by a physical network. The signal is then converted to a digital form which can then be tuned and processed with by the computer. The ADC samples the signal at a rate of 1000 Hz. Following figure shows us the parameters and block configuration of our Converter:



Our sampling rate corresponds with the Cut-off Frequency of our Sallen-Key Filter, f_c = 500 Hz, and is responsible for not capturing the 1100 Hz AC Source.

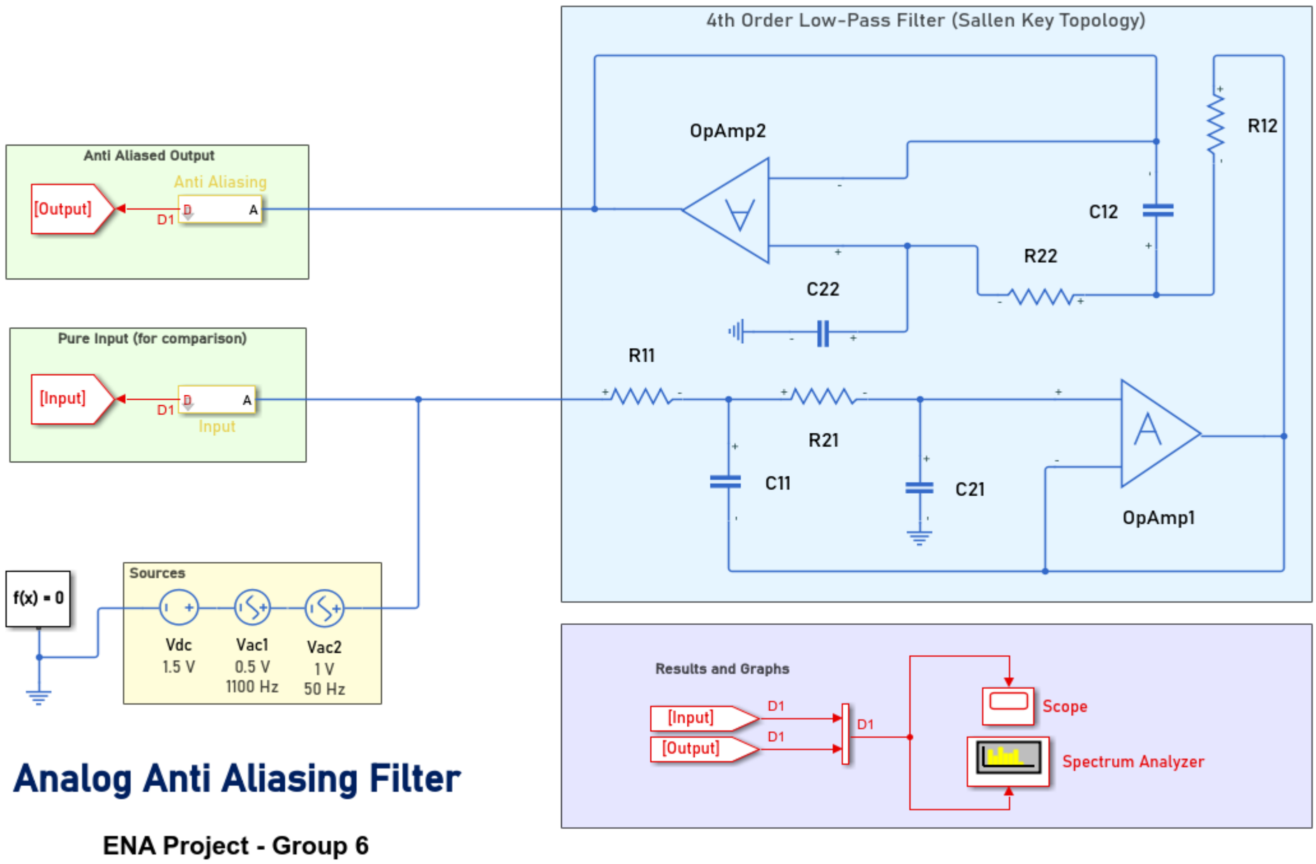
There are two GOTO – FROM to redirect the signal back to our last segment which consists of the Scope and Spectrum Analyzer. These are responsible for all the simulation of both the Input and Output signals. It should also be noted that the model at hand, Analog Anti-Aliasing Filter, is made to run for only 2 seconds. Number of bits correspond to the number of steps that we've already covered in the aforementioned section; A-D Converters.



Simulation Area

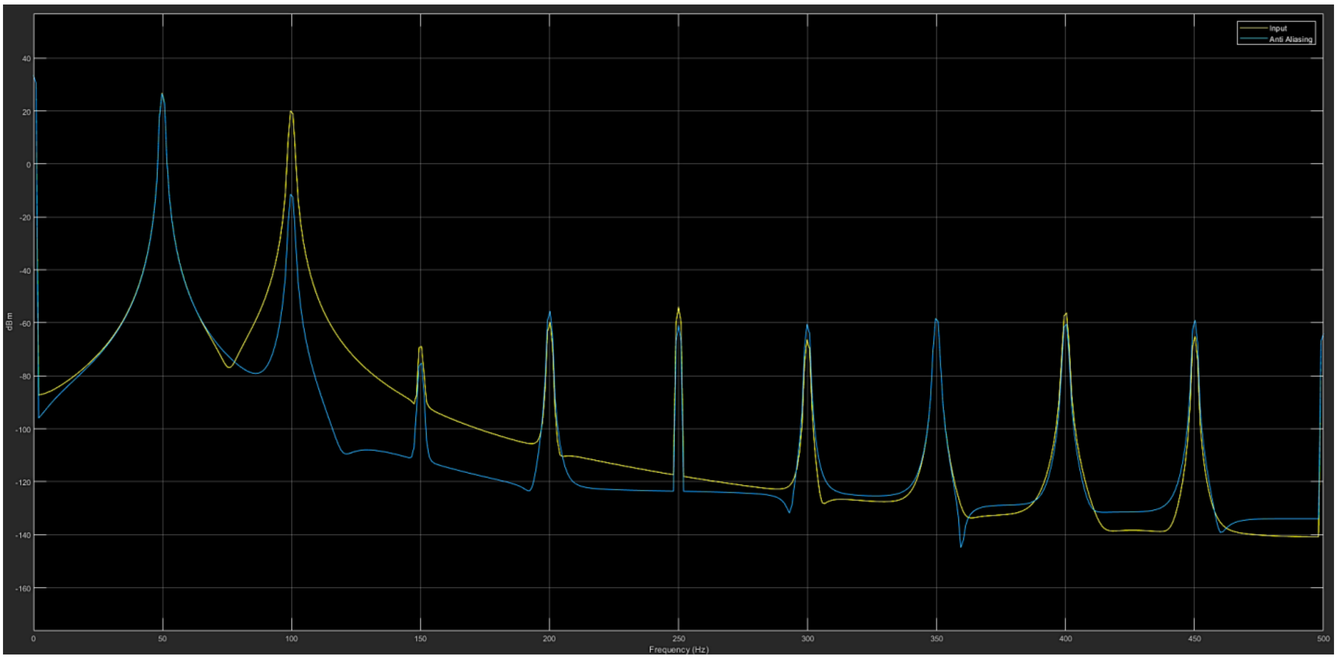
The ‘Vector Concatenate’ component here performs the jobs of creating a vector input of our output that is then separated into two streams. It has no impact on the Signal itself as it is purely for formatting purposes, i.e. we would have to use two of both the Scope and Spectrum Analyzer had we not used the Concatenating component.

Combining all these sections give us our complete model as shown below:

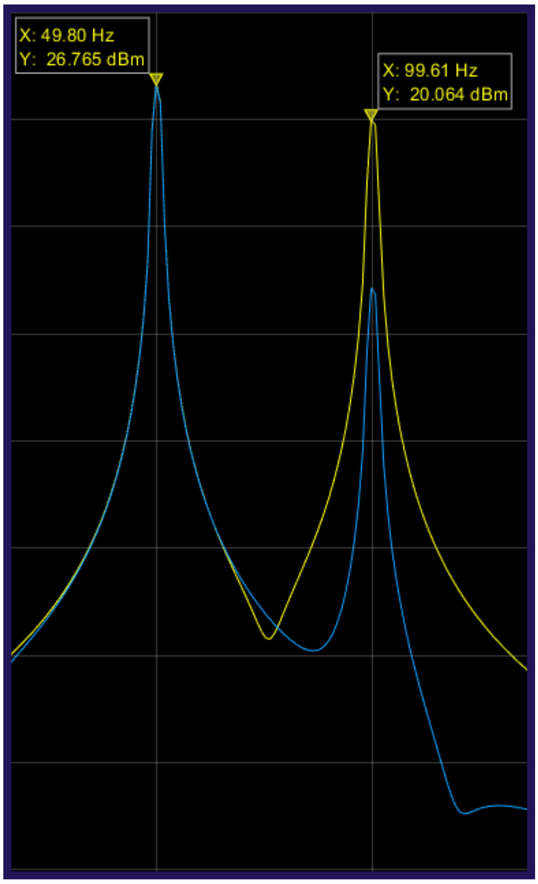


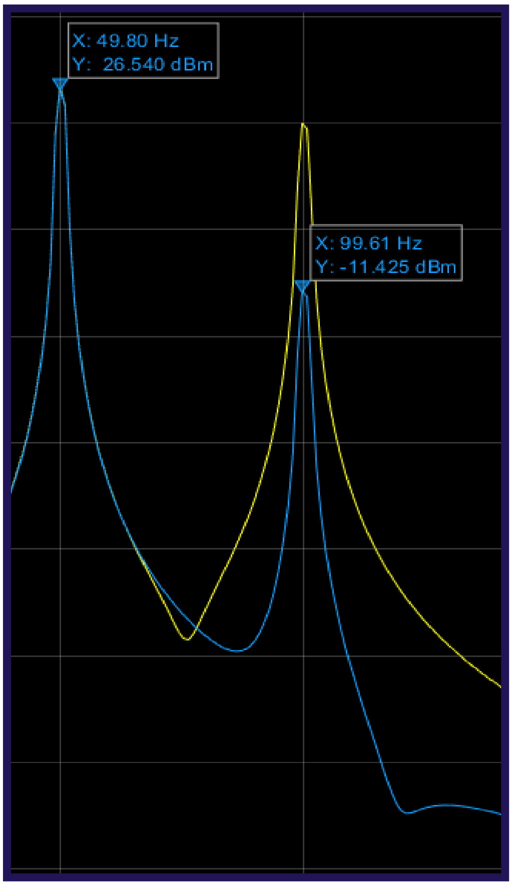
Combining this with the following ‘Stop Time’ gives us the ability to proceed to the final section:

Results



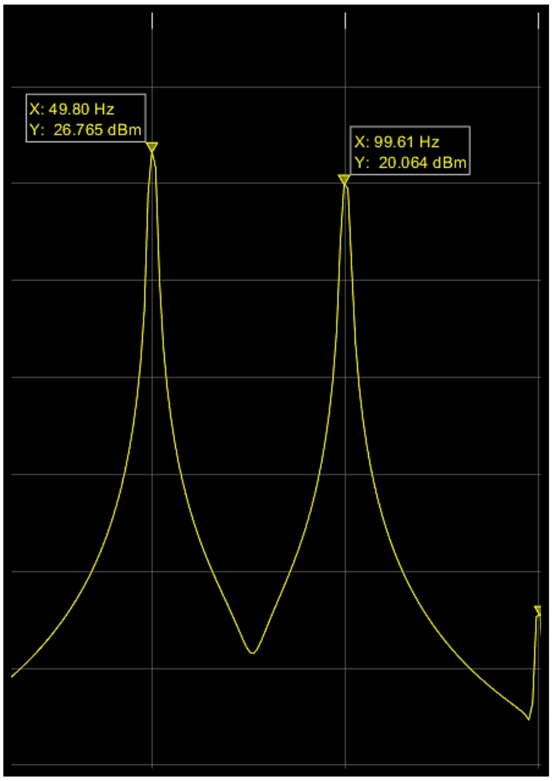
This figure portrays the Output from our Spectrum Analyzer with a reference load of 1 Ohm. Notice how the second harmonic in the blue signal, which is the one passed through the Anti-Aliasing filter, is 37 dB down compared to our Input signal.

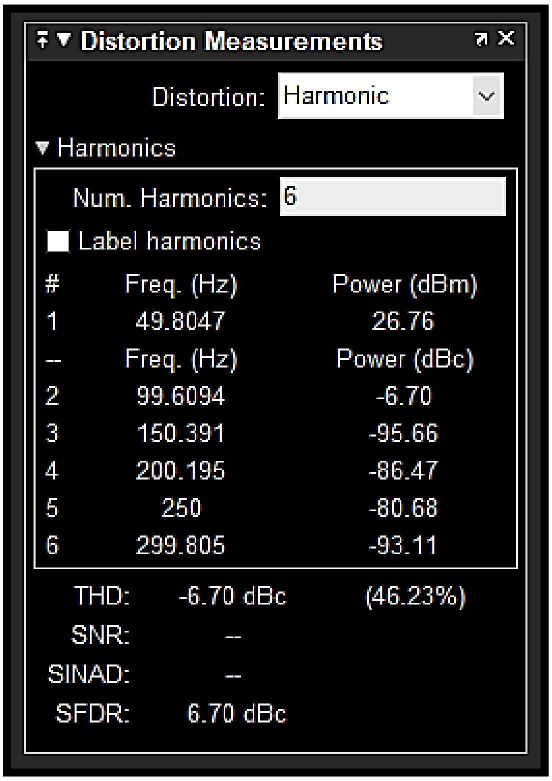




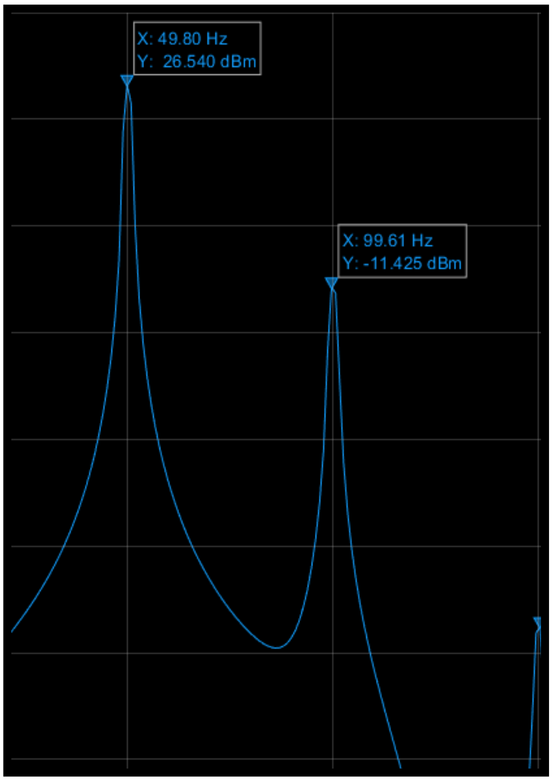
Had it been a Variable-Frequency Oscillator, our Input here would clearly not meet the emission requirements established by reputable organizations.

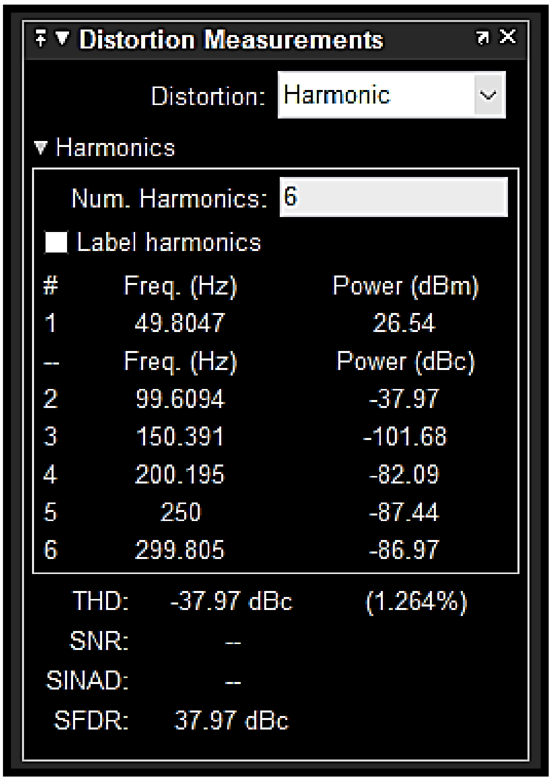
Enabling Peak Finder and Distortion Measurements from Tools, we can find the Total Harmonic Distortion in both our Input and Filtered Signals. Since we are only concerned with relative distortion, observing only the fundamental and 2^{nd} harmonic would suffice.





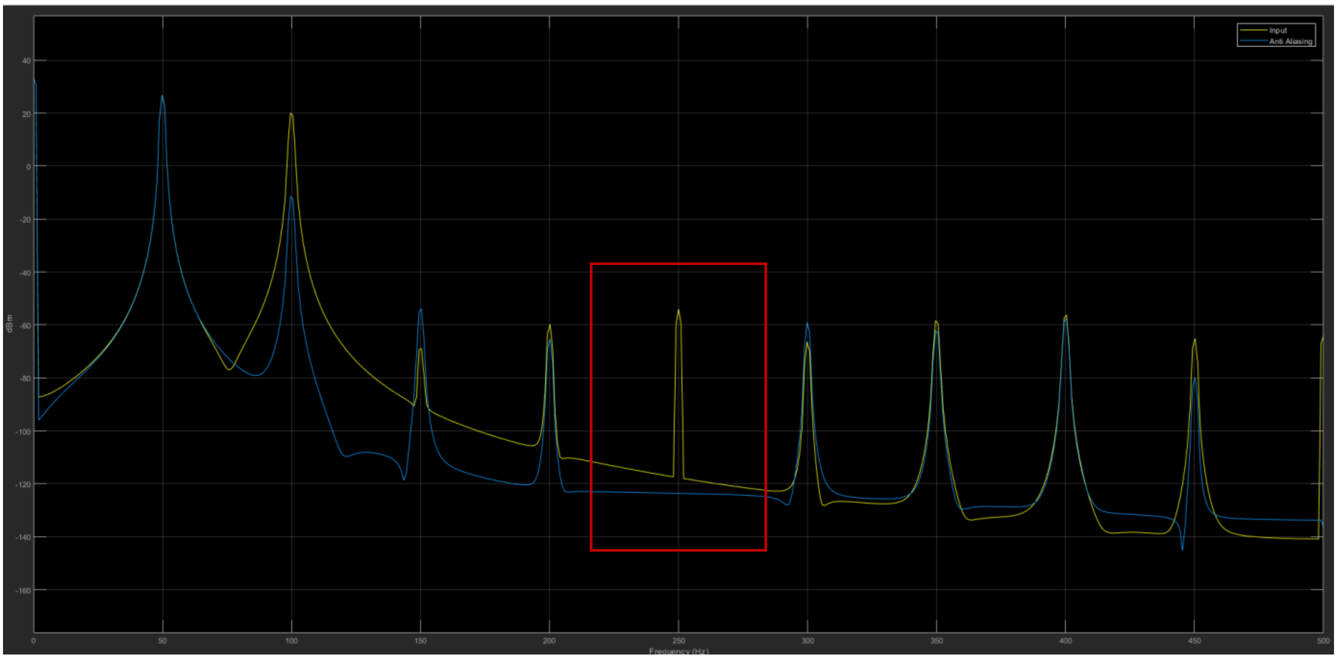
Which displays our Input signal's fundamental harmonic and the 2^{nd} harmonic; And such amount of dB down corresponds to 46.23% of Total Harmonic Distortion. Now, looking at our filtered signal:





We can observe that there has been a decrease in the percentage distortion by almost 45% which our Anti-Aliasing filter is supposed to do.

It also to be noted that had we used a single Op Amp, i.e. a 2nd Order LPF, we would miss a harmonic implying that our original signal had been altered more than just the desired removal of aliasing.



The following section demonstrates and explains the script linked with the Simulink model. At its core, it logs all the data on the output of our implemented model. The workspace variable of Simscape Logging is coined under the term **'simlog_aafilter'**, from which we derive our series time, the voltage input and the voltage output.

```
% Script linked with Simulink Project

% Generates new simulation results
sim('aafilter')          %% displays the spectrum analyzer
```

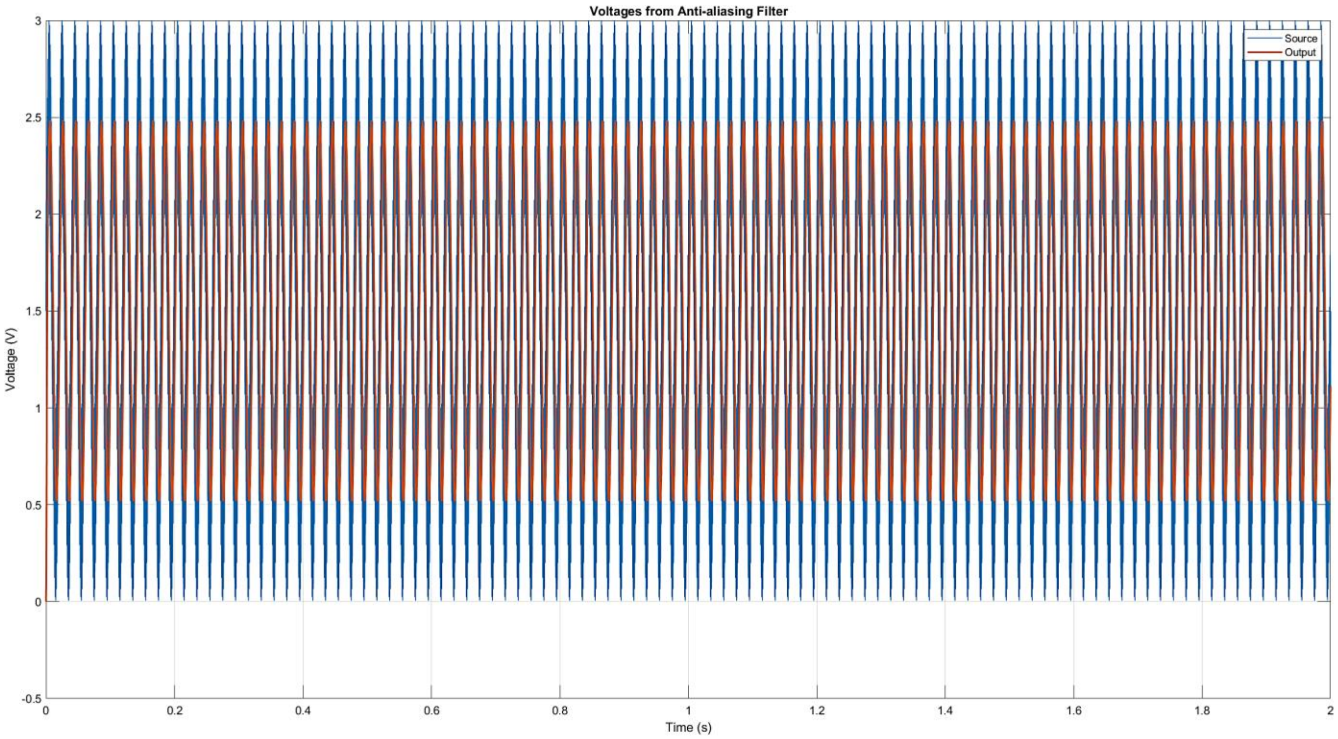
- MATLAB will automatically look for the Simulink model with the file name "aafilter" in the parent folder. If found, it is equivalent to running the simulation itself

```
% Get plot definitions
simlog_t = simlog_aafilter.Vac2.p.v.series.time;
simlog_Vin = simlog_aafilter.Vac2.p.v.series.values('V');
simlog_Vo = simlog_aafilter.OpAmp2.out.v.series.values('V');
```

- Derives/Extracts the Time, Input Voltage over the Vac2 and the Output Voltage after OpAmp2 from the Simscape Logging Workspace and stores them into separate variables with names that we can then use to plot the I/O Voltages versus Time

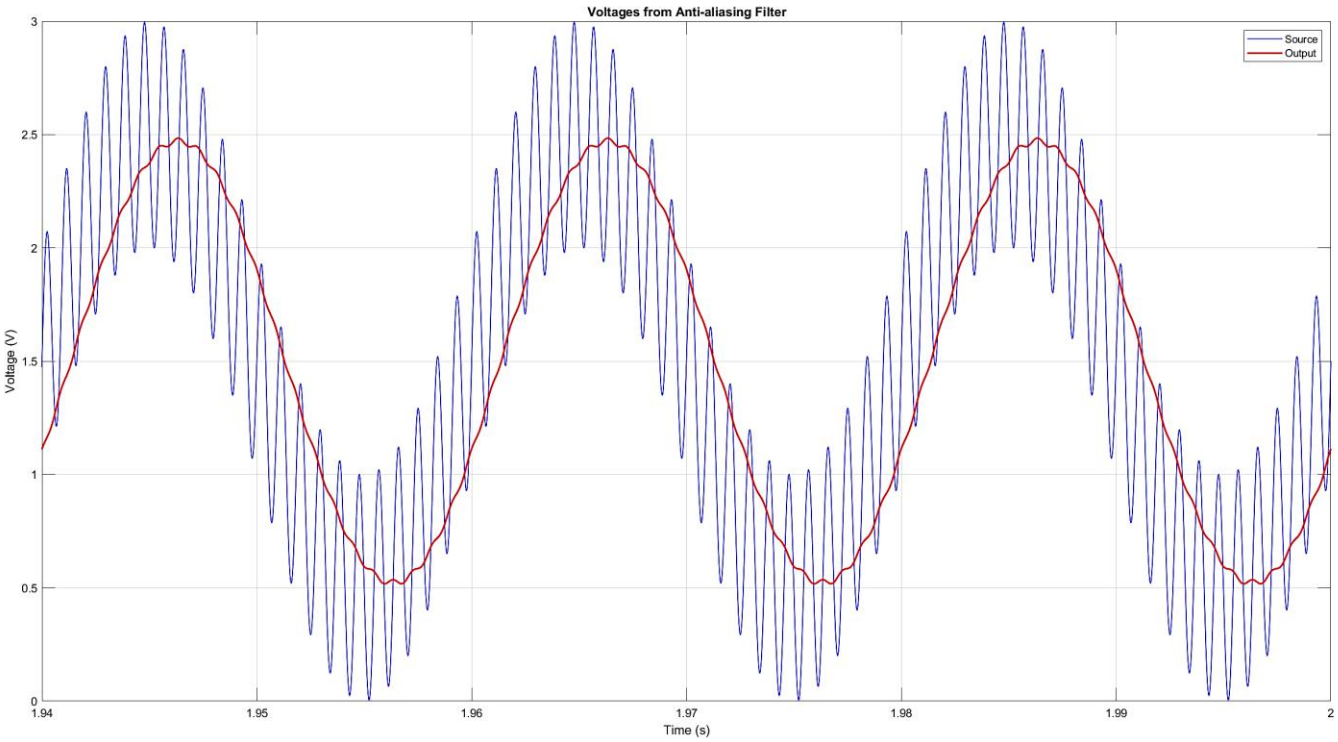
```
% Plot config and display
plot(simlog_t, simlog_Vin, 'LineWidth', 0.75)
hold on
plot(simlog_t, simlog_Vo, 'LineWidth', 1.5)
hold off
grid on

title('Voltages from Anti-aliasing Filter')
ylabel('Voltage (V)')
legend({'Source','Output'});
xlabel('Time (s)')
```



However, since we only are concerned with comparing the voltage signal of the input with that of the filter on (OpAmp2) and not the whole signal itself, we can impose limits on our X-Axis, i.e., limit the interval of time that is displayed by the scope. This is best done by using the GCA handle and the built-in Set command. The additional script line is as the following:

```
set(gca,'XLim',[simlog_t(end)-0.06 simlog_t(end)]);   % Sets the limit for starting
                                                         and ending points of X axis
```



The red signal displays the output signal from the filter, displaying the correct magnitude of 1 Volts of the desired 50 Hz sine wave.

VIII. Conclusion

In this Complex engineering problem, we removed the distortion in our signal from about 46% to less than two percent using a second order two stage active filter with Sallen-Key topology and then passing it through the anti-aliasing filter. Our first input signal that was seen through the scope was not passed through the Anti-Aliasing filer and because its frequency was greater than the sampling rate Our output signal was not matching our readings. Next, we passed the other part of our input signal through an active low pass filter with a non-inverting amplifier whose purpose was to attenuate the input signal through the cut off frequency of 500Hz. When the signal was attenuated the frequency of the signal reduced to 500Hz. Now this reduced frequency was passed through the analogue to digital converter which had a sampling rate of 1000Hz which is double than the cut off frequency. Then this signal was passed through the anti-aliasing filter to remove the distortion in our signal.

In conclusion, we were able to learn the different uses of Anti-Aliasing filter and how to remove distortion in our input signals. These filters are used in gaming and graphics to reduce the noise in our reading and to bring efficiency on our results. Moreover, aliasing is an efficient and economical method to remove most kinds of distortions from our input signal. Our project consisted of MATLAB code and a Simulink profile which demonstrated in an effective manner.