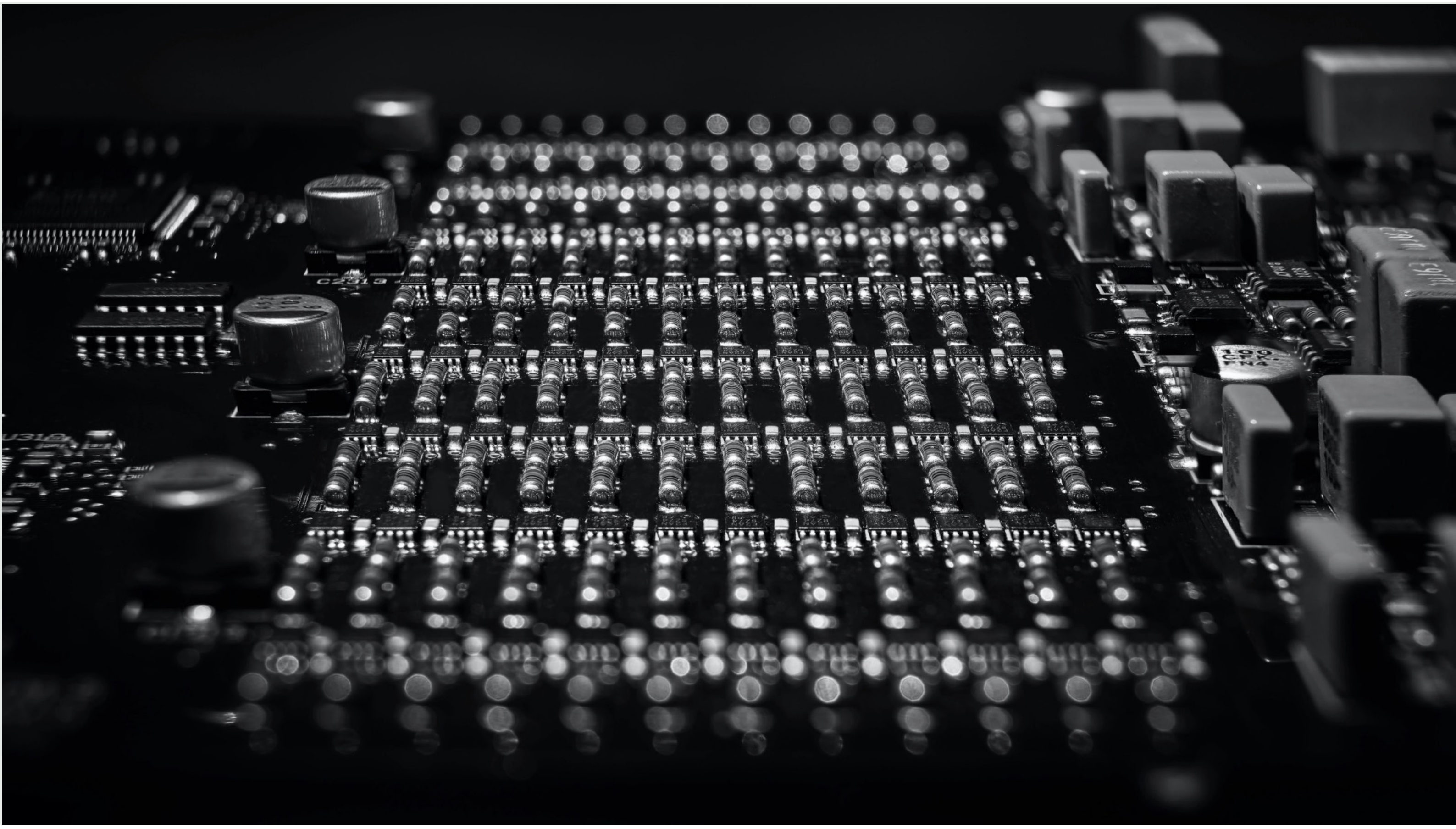
dCS
ONLY THE MUSIC

Filtering in digital audio

Exploring why filters are needed, how they work, and the dCS approach to filtering



Synopsis

Most Digital to Analogue Converters (DACs) will have some information in their specifications about the types of filtering they use. As these filters are an incredibly important part of the product, it is worthwhile explaining why and how they are used.

In this paper, we'll examine how filters work in both DACs and Analogue to Digital Converters (ADCs), providing an overview of their role, importance and impact on digital playback. We'll discuss common filter types, such as low-pass, half-band, and asymmetrical, and show how different filter shapes have different effects on sound, as well as the different filtering requirements of DACs and ADCs. We'll also examine dCS's approach to filtering and our reasons for providing a range of specially designed filters for users to select from within each dCS DAC.

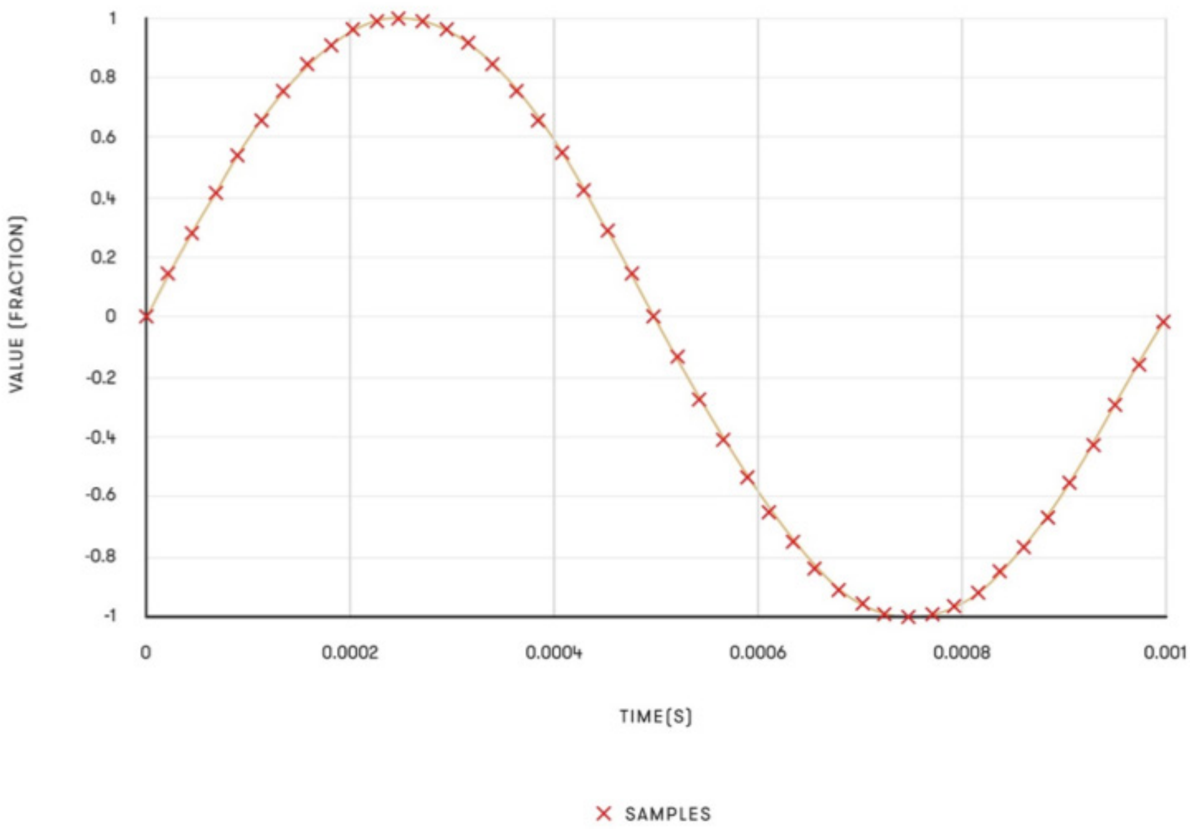
Why we need filters, Part I: A/D conversion

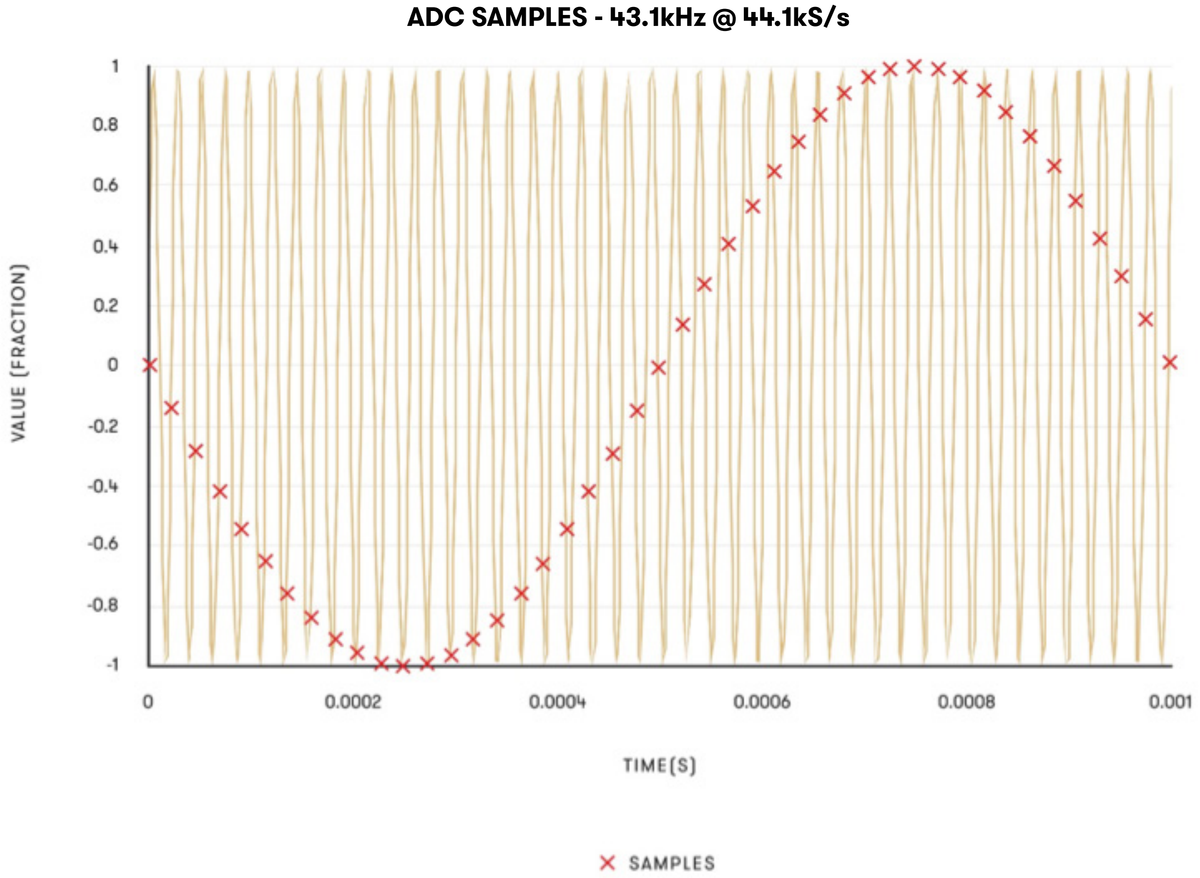
To understand why a filter is needed, it helps to start at the beginning, when an analogue signal enters an ADC during the recording / production process. (This is significant as the filter within an ADC has almost as much impact on what we hear during playback as the filter within a DAC.)

Our paper on the dCS Ring DAC explored how audio is sampled using an ADC. (The analogue voltage is converted into a digital representation, with a series of 'samples' being taken to form this representation.) The lowest sample rate used in audio is typically 44,100 samples per second (S/s). The reason for using this sample rate (44.1kS/s) is largely due to the Nyquist Theorem. This states that the sample frequency of digital audio needs to be at least twice the highest frequency in the audio being sampled. The highest frequency which can be sampled (half of the sample rate) is defined as the 'Nyquist frequency'. As the human range of hearing extends up to 20,000Hz, accurately sampling this frequency range requires a sample rate of at least 40,000S/s.

However, what happens if what we are sampling doesn't 'fit' into our sample rate's valid range, between 0Hz and the Nyquist frequency? If this occurs, then the frequency components above the Nyquist frequency are 'aliased' down below it. This sounds counterintuitive, but it is illustrated here:

ADC SAMPLES - 1kHz @ 44.1kS/s

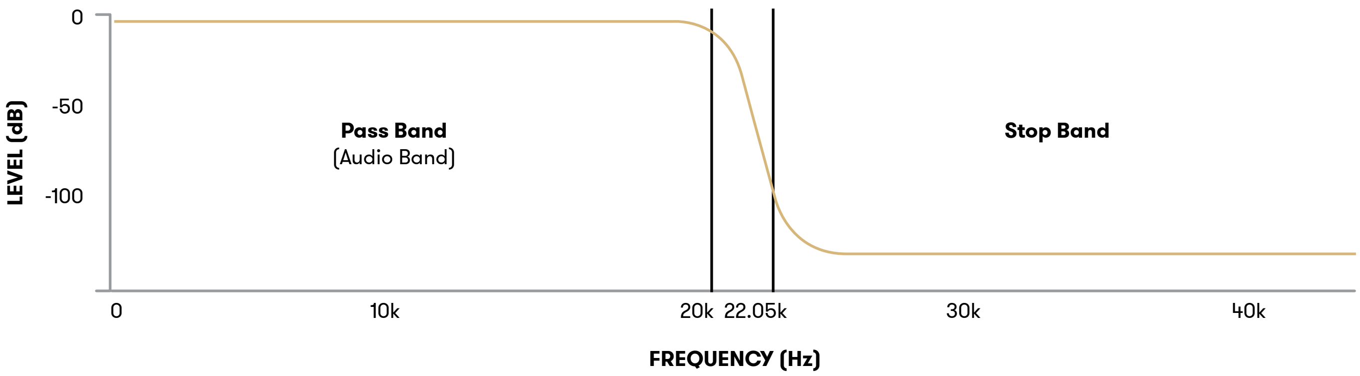




The above graphs show two signals: one at 1kHz and one at 43.1kHz, both sampled at 44,100 samples per second (44.1kS/s). Note that sampling the 43.1kHz signal produces samples which are indistinguishable from the 1kHz tone (though phase inverted). If this 43.1kHz signal was passed through the ADC, the resultant samples would be indistinguishable from those of the 1kHz tone – and a 1kHz tone would be heard on playback. This means that the ADC must remove anything which does not 'fit' between 0Hz and Nyquist frequency, to avoid these aliased images affecting the audio.

The removal of anything which does not fit between 0Hz and Nyquist frequency is carried out by way of a low-pass filter. This filter removes any content above a certain frequency, and allows anything below that frequency to pass through, ideally unchanged. This filter can be implemented in either the digital or analogue realm.

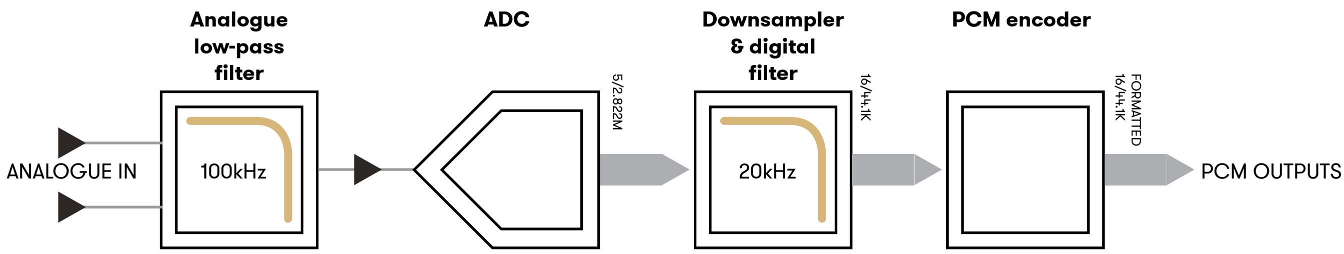
It would seem that the most obvious solution to the aliasing problem caused by running an ADC at a sample rate of 44.1kHz is to implement a filter which does nothing at 20,000Hz, but cuts everything above 20,001Hz. This would allow the removal of any unwanted alias images from the A/D conversion, while ensuring the audio band remains unaffected. However, such a filter is highly inadvisable. For one thing, if using a digital filter, the computing power required to run such a filter would be excessive. Filters work by reducing the amplitude of the signal above a frequency on a slope so to speak, measured in decibels per octave. As such, the audio is sampled at a higher rate than simply double the highest frequency we are trying to record (it is actually sampled at 44,100Hz instead of at 40,000Hz), which allows some room to filter it. This means the filter can now work between 20,000Hz and 22,050Hz without aliasing becoming an issue, while also leaving the audio frequencies humans can hear unaffected.



This diagram illustrates a low-pass filter for 44.1kHz audio.

This is still an extremely narrow 'transition band' to play with. If this is done with an analogue filter, the filter will have to be very steep – this is problematic as analogue filters aren't phase linear (the filter will delay certain frequencies more than others, causing audible issues) and are pretty much guaranteed to not be identical. This is okay when they are working at say, 100kHz, but at 20kHz this becomes very problematic. As such, the filter used to remove any content from the Nyquist frequency and up is implemented in the digital domain, in DSP (Digital Signal Processing).

In audio recording, it is common practice to use a high sample rate ADC and perform the filtering at the Nyquist frequency on the digital data instead. This method is known as an 'Oversampling ADC'. The block diagram for a dCS oversampling ADC producing 16-bit 44.1k data is shown here:



The analogue low-pass filter removes high frequencies from the analogue signal above 100kHz, as these would cause aliasing. As previously discussed, this analogue filter acting at 100kHz can be gentle and acts in a region where non-linearities are not as critical.

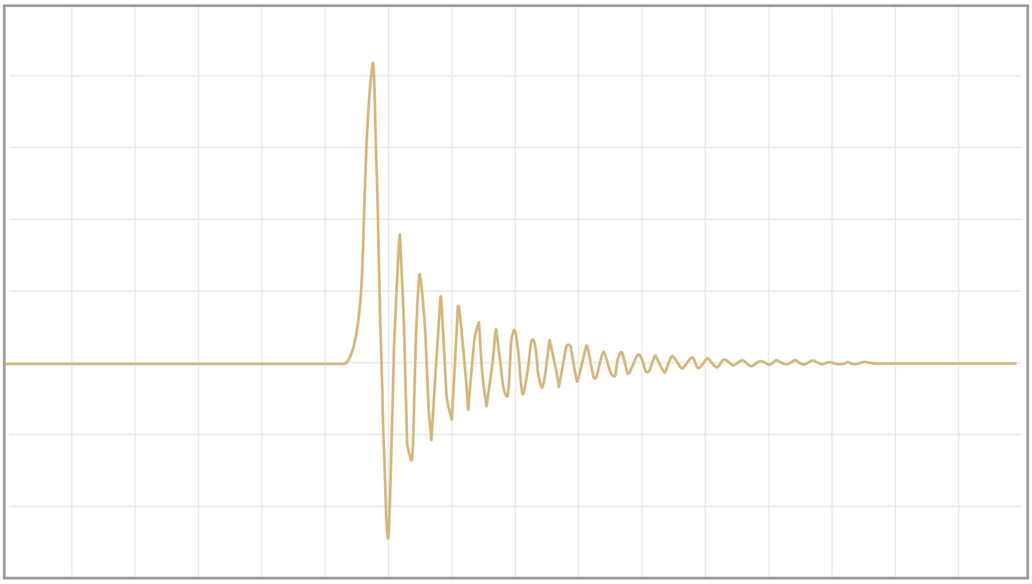
The ADC stage then converts the signal to high-speed digital data. In a dCS ADC, this stage is a Ring DAC in a feedback loop, so produces 5-bit data sampled at 2,822,000 samples per second.

The Downsampler converts the digital data to 16-bit, 44,100 samples per second. This data then passes through a sharp digital filter, which effectively removes content above 22.05kHz. (Frequencies higher than this will cause aliases if not filtered out.) The PCM encoder then formats the data into standard SPDIF, AES/EBU and SDIF-2 serial formats, complete with status and message information.

The digital filter used in the Downsampler will have its own set of trade-offs to employ. To simplify this greatly, digital filters work by passing each sample through a series of multipliers, with these multipliers collectively acting to filter higher frequencies from the signal. The shape of how these multipliers are arranged is referred to as the filter 'shape' (symmetrical or 'half-band' filters, asymmetrical filters). Different filter shapes have different impacts on the sound.

This diagram illustrates an example of the response of a symmetrical digital filter. They are called this as they produce symmetrical 'ringing' when driven with an impulse (also known as a transient). This results in an acausal response before the impulse. The effect is more pronounced at lower sample rates:

This diagram illustrates an example of an asymmetrical filter response. This filter type has a completely different impulse response – here, there is no ringing before the impulse, but there is more ringing after the impulse when compared to a symmetrical filter:



Given the fact that the ADC must use a filter to remove aliases, and that a digital filter acting at the Nyquist frequency is preferable to using a harsh analogue filter, there will therefore be pre- and/or post-ringing introduced at the recording stage by the digital filtering in the ADC. This is a good trade-off to make, and the filter choice here is important.

Most ADCs will work using a symmetrical filter. What this means is that for any digital recording, there will be (necessary) pre- and post-ringing present on the recording, as a result of the filter which was used. The key point to be made here is that all digital recordings will include ringing from the filters, even before they reach the DAC, but this is the best approach to take – provided the filters are correctly designed and implemented within the ADC.

Why we need filters, Part II: D/A conversion

The other side of this topic is the DAC, where the digital audio recorded by the ADC is translated back to analogue for playback.

When a DAC reproduces an analogue waveform from digital samples, an effect similar to aliasing occurs. This is where, due to the relationship between the frequency of the analogue audio signal and the sample rate of the digital signal, 'copies' of the audio spectrum being converted can be observed higher up in the audio spectrum. While these images exist at frequencies outside the range of human hearing, their presence can have a negative impact on sound.

There are two reasons for this. Firstly, frequencies at rates above 20,000Hz can still interact with and have an audible impact on frequencies lower down, in the audible spectrum (between 0-20,000Hz).

Secondly, if these images – known as Nyquist images – are not removed from an audio signal, then the equipment in an audio system may try and reproduce these higher frequencies, which would put additional pressure on that system's transducers (particularly those responsible for reproducing high frequencies) and amplifiers. Removing Nyquist images means an amplifier has more power available to use for reproducing the parts of an audio signal that we do want to hear, which leads to better performance and a direct positive impact on sound.

> "The question of how a low-pass filter should be designed is a complex and sensitive topic, and it's important to note that there is no one-size-fits-all solution."

Similar to in an ADC, the solution to the problem posed by Nyquist images in D/A conversion is to filter anything above the highest desired frequency of the audio signal by using a low-pass filter. This allows Nyquist images to be eliminated from the audio signal, without impacting the music we want to hear. The question of how a low-pass filter should be designed is a complex and sensitive topic, and it's important to note that there is no one-size-fits-all solution.

Of course, when working with source material which is at higher sample rates than 44.1kHz, such as hi-res streamed audio, the requirements of the filter in the DAC change. There is a naturally wider transition band and as such, the filter requirements are different. Most DAC manufacturers offer a single set of filters which are cascaded for different sample rates. Given the different filtering requirements posed by converting different sample rates, this is not the optimal approach to take in a high-end audio system.

For this reason, the filters found within dCS products and the Ring DAC are written specifically for each sample frequency by dCS engineers. Further to this, there are multiple filter choices available for each sample frequency in a dCS product. There is no one right answer to filtering, as it depends on the listener's preference and the audio being reproduced, so a choice of very high-quality filters bespoke for the Ring DAC and the sample frequency of the audio are available for the user to choose from.

Filter design: understanding the requirements of DACs and ADCs

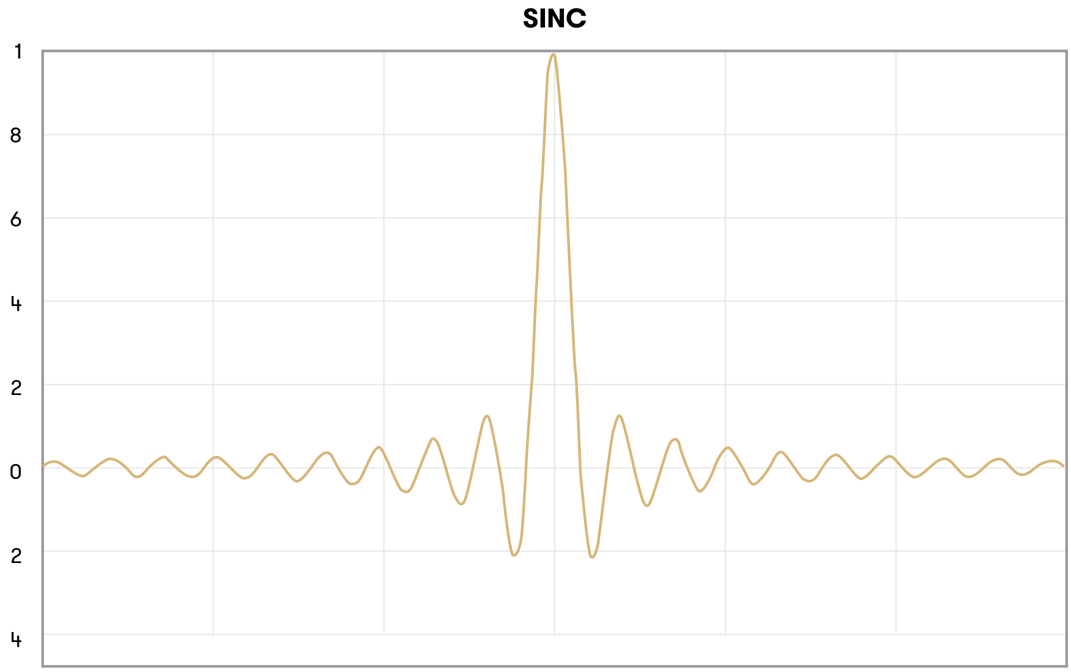
How do the digital audio filters discussed above work, and how should they be designed? This may sound counterintuitive, but the filtering required in both ADCs and DACs is actually very similar.

In an ADC, the audio is coming into the converter at a higher rate than we want to output – typically by a power of 2. To deal with this, the converter must remove content above the Nyquist frequency, which allows it to drop samples (allowing it to lower the sample rate from 88.2k to 44.1k for example) without content from above the Nyquist frequency aliasing down below it. This is handled in Digital Signal Processing by way of low-pass filtering.

In an oversampling DAC, audio is coming in at a lower rate than we want to feed to the converter. There are several approaches for how to tackle this issue, but by far the most effective is to insert samples with an amplitude of zero between the actual audio samples to increase the sample rate, then low-pass filter the signal to remove the Nyquist images this process creates. Once again, DSP is used to implement low-pass filtering.

Co-efficients, taps and the sinc function

How is this digital low-pass filtering carried out? In simplified terms, the digital audio signal is run through a series of 'coefficients' – multipliers which change the amplitude of the audio sample by an amount, defined by a number between 0 (no output) and 1 (the original full amplitude of the sample). Each of these coefficients is what is referred to as a 'tap'. A higher number of taps means the signal is run through a larger amount of coefficients in the filter. The output of the filter at any one point is the sum of all of these coefficients multiplied by the respective samples.



Audio websites and magazines often feature 'impulse response' plots of filters in audio products. These typically show the output from a filter when it is given samples, all with an amplitude of zero, then a single full-scale (all 1s) sample, then all zero samples again. The effect of this is to show the coefficients in a filter, which define how the filter works. Typically, this filter is a derivative of what is known as the 'sinc' function. The sinc function is defined as sin(x)/x and looks a little like the below graph.

There are several useful properties of the sinc function – it acts as an ideal low pass filter (the set of coefficients used in a digital filter, as previously described, are taken from this function, hence the Y axis being shown as ≤1, which will be shown below); it is mirrored in time in both directions (before and after the impulse, the single full-scale sample shown in the middle of the above sinc graph) and it offers the same delay at all frequencies (not all filters do this). An analogue filter will delay some frequencies more than others, creating phase issues. Filters which offer the same delay at all frequencies are referred to as phase linear.

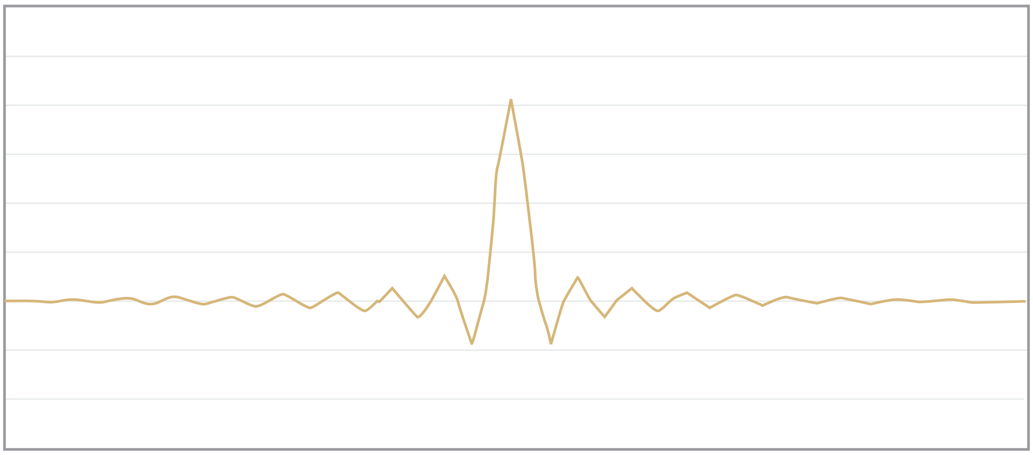
Given these factors, the sinc function can be manipulated to provide the desired frequency response. The three main factors that can be adjusted with digital filters are:

- The -6dB point – the frequency at which the filter reaches 6dB of attenuation.
- The filter length – the number of coefficients used in the filter, with one coefficient often being referred to as a tap'.
- The windowing technique – this is linked to both of the above factors (this is a simplified explanation)

Cutoff Frequency (-6dB Point)

To explore the factor of the cutoff frequency of a filter, take the below graph, which shows the most commonly used digital filter. This is a 'half-band' or 'Nyquist' filter, with coefficients based off of the sinc function. This filter is designed in such a way that the -6dB point is at the Nyquist frequency of our target. For a low-pass filter in, for example, an ADC, the Nyquist frequency is set at 22.05kHz. For a tap length of 128, this generates the below impulse response.

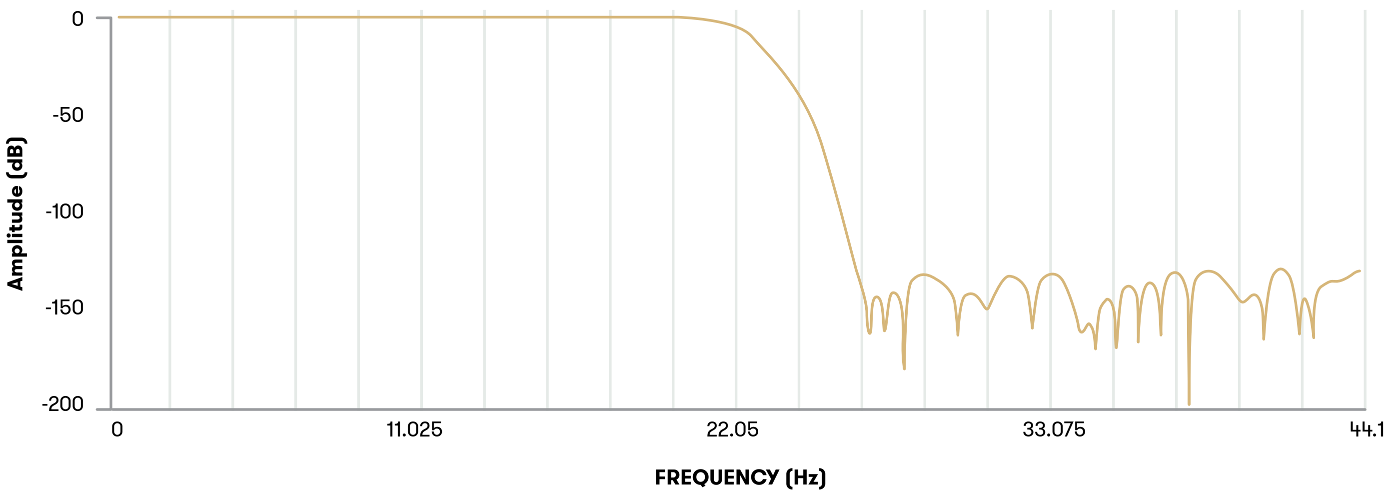
HALF-BAND FILTER IMPULSE RESPONSE



The crucial aspect of this filter is that every second coefficient is 0. This makes it twice as computationally effective as a non-Nyquist filter, as using a coefficient of 0 always results in an output of 0.

The drawback to this type of filter is that, by definition, it only reaches 6dB attenuation at the Nyquist frequency. This means that aliasing artifacts from the ADC mirrored down to below Nyquist frequency will not be correctly attenuated. We would therefore ideally want the -6dB point to be **below** the Nyquist frequency, to remove said aliases and maintain good attenuation at the Nyquist frequency.

HALF-BAND FILTER FREQUENCY RESPONSE



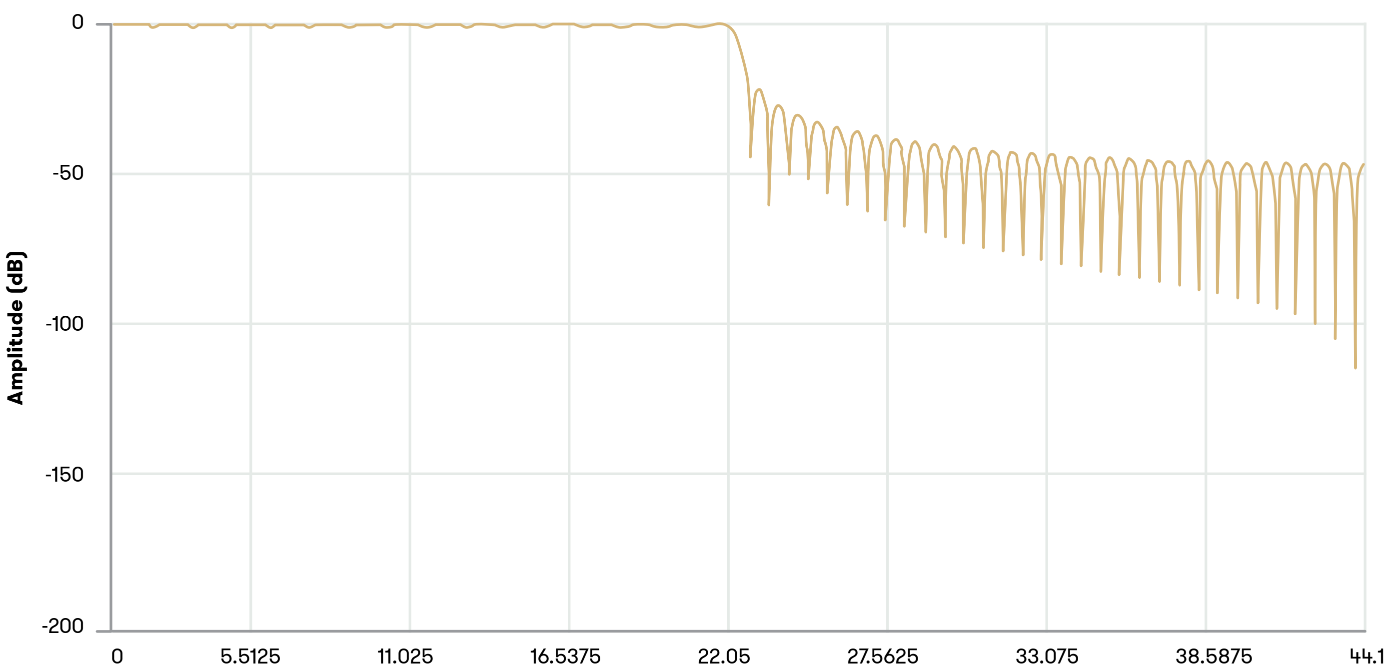
Windowing Technique

There are some drawbacks to using the sinc function as the basis for a digital filter – firstly, the filter would be infinitely long (due to the sinc function being a mathematical function with no defined length), which is problematic in the real world. It also requires data from the future, as the filter coefficients include data from *before* the current sample being played. So how should these issues be addressed?

A good start would be to define the filter as *not* infinitely long. Doing so would allow the incoming audio signal to be buffered by a certain number of samples. This means the filter can then effectively have some data from the 'future' by delaying the entire audio signal by that amount of samples.

However, defining the filter with a finite length leads to some mathematical problems – the sinc function starts to get very small when moving further away from the central impulse, but not so small that the function becomes irrelevant. Sinc is infinitely long, but it is unwise to simply select a finite section of the sinc function to use as the basis for a filter, chopping off the 'start' and 'end' of the function to create a finite length. Doing so actually causes the resultant filter to not work very well. The below graph shows an example of a digital low-pass filter designed to work for CD rate audio, with a tap-length of 64 taps, where the ends of the sinc function have simply been chopped off to provide the coefficients used in the filter.

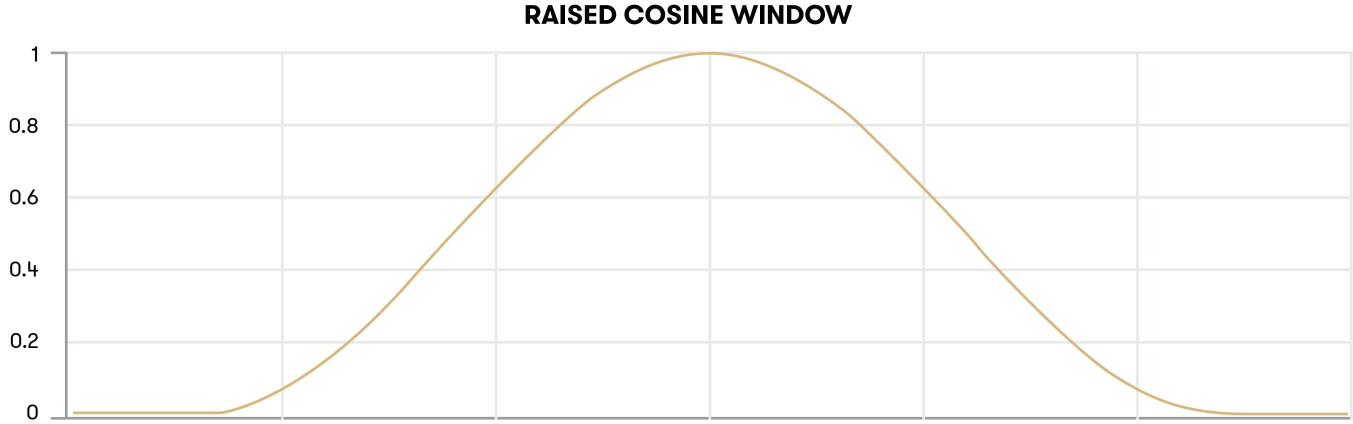
64 TAP FILTER FREQUENCY RESPONSE, NO WINDOW



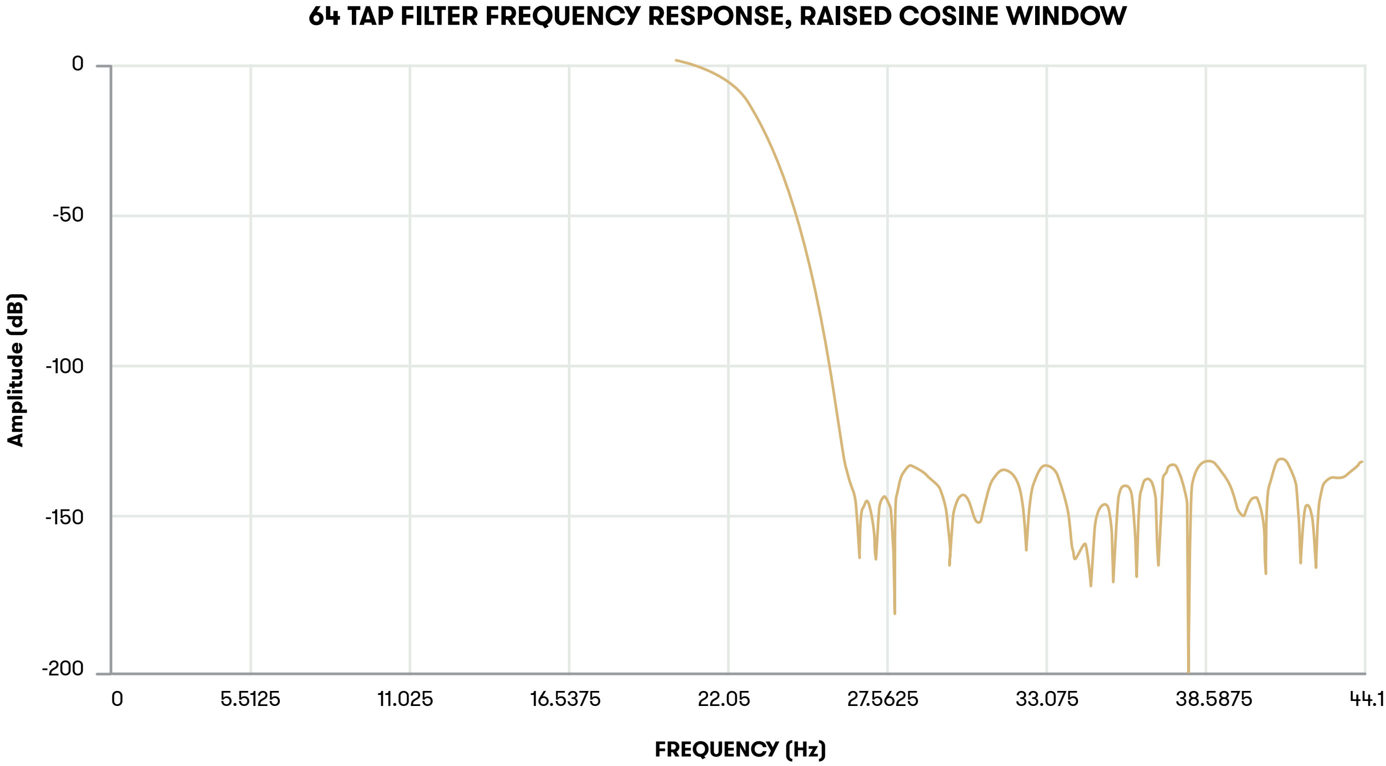
FREQUENCY (Hz)

As can be seen from this graph, this filter does not provide a good response. The stop-band rejection above 22.05kHz is poor, and there is a significant ripple in the passband (below 22.05kHz). This problem is best addressed through a technique called 'windowing'. This involves taking the coefficients for the filter we want to use, such as a finite section of the sinc function, and then multiplying them by another set of coefficients to reduce the negative effects seen above.

There are many approaches to windowing (different sets of functions / coefficients) used, but for this example, a raised cosine window will be used.



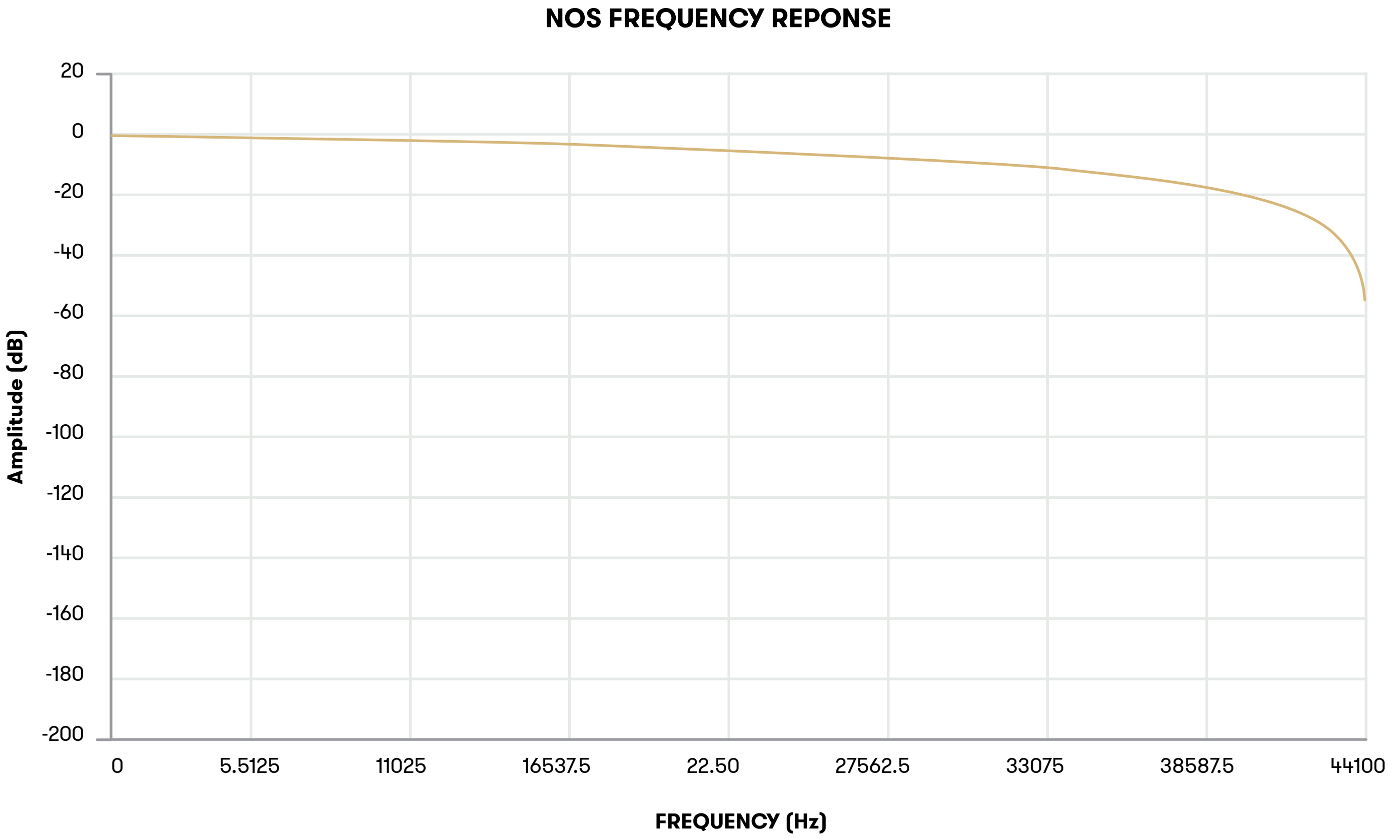
Applying this window to the above 64-tap filter results in the below frequency response:



As can be seen, this provides drastically better results. The stopband has far better rejection and the rippling seen in the passband is no longer present. What this means is that when designing a filter for use in audio, the windowing function needs to be selected carefully for correct filter performance.

Filter Length

The third factor to consider with filter design is the length of the filter itself. As previously discussed, we need to filter the output of a DAC to prevent imaging. This filter needs to be a finite length – longer than nothing but not infinitely long. If this is not done and no filtering is carried out to the DAC's output, and the samples are played back as they come in, the result is not ideal:

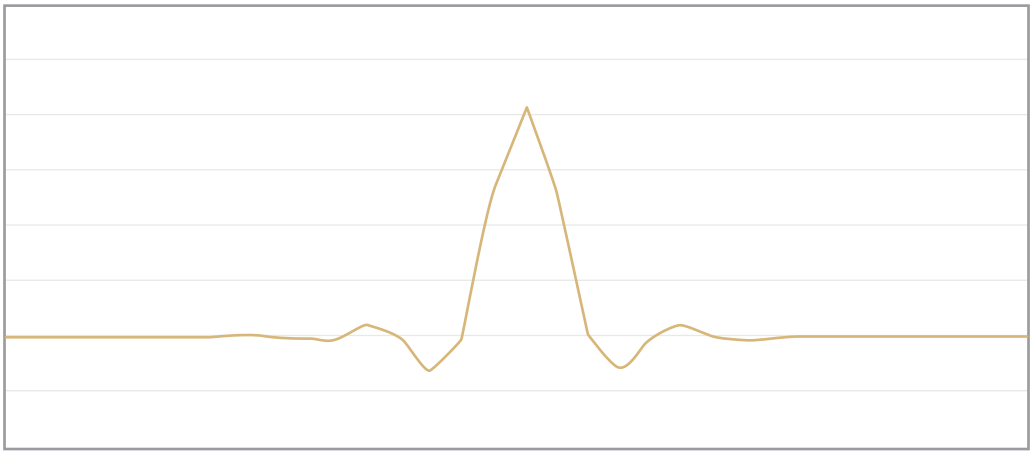


This graph shows the frequency response of a Non-Oversampling (NOS) DAC with no filter on the output.

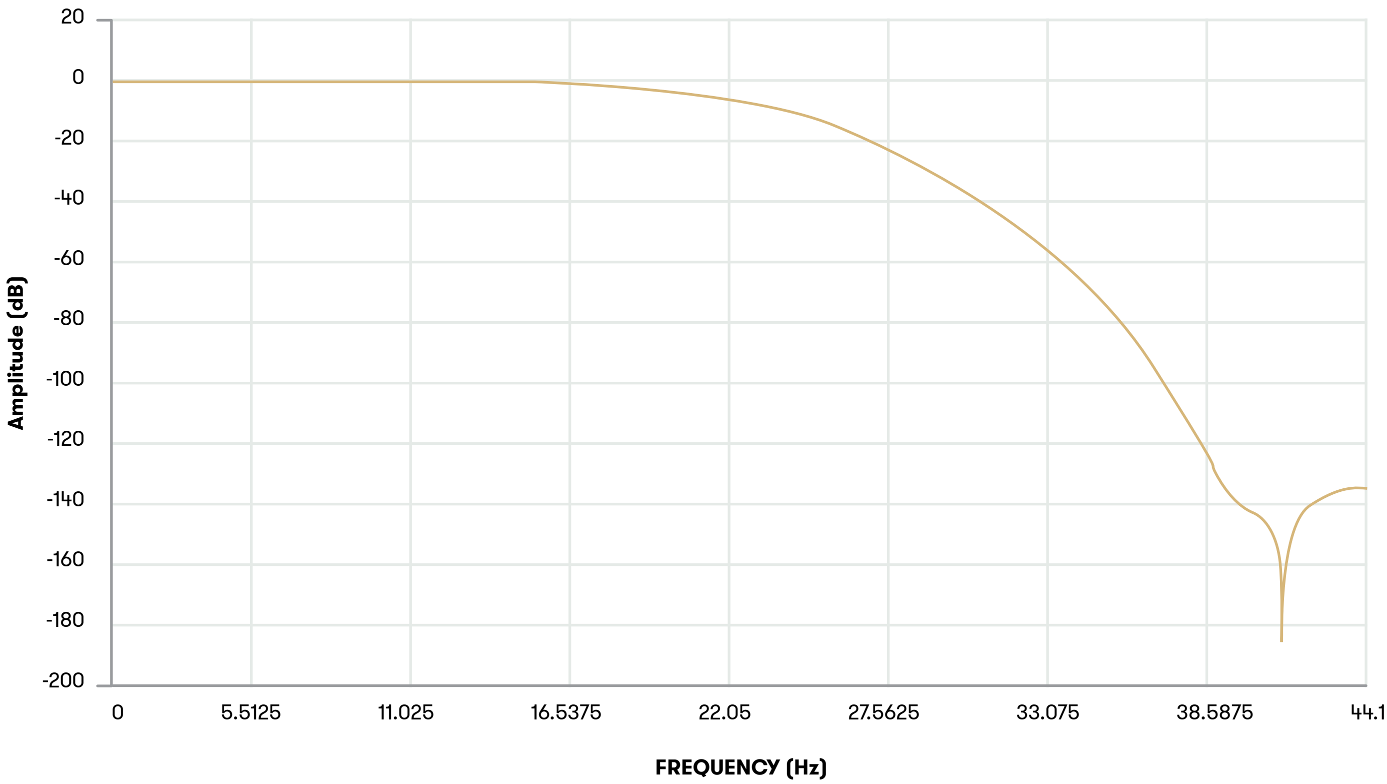
As discussed previously, a filter for 44.1k digital audio should not affect signals below 22.05kHz, but should heavily attenuate them above this. With no filter in place, not only is there very little attenuation above 22.05kHz (thus leading to lots of false high frequency components), we are also affecting the signal we are interested in: the frequency response at 20kHz is -3dB.

So, how long should the finite length filter on the DAC output be? To illustrate this factor, the same previously mentioned Nyquist filter with a raised cosine window will be used. Firstly, a 32 tap filter:

32 TAP FILTER IMPULSE RESPONSE



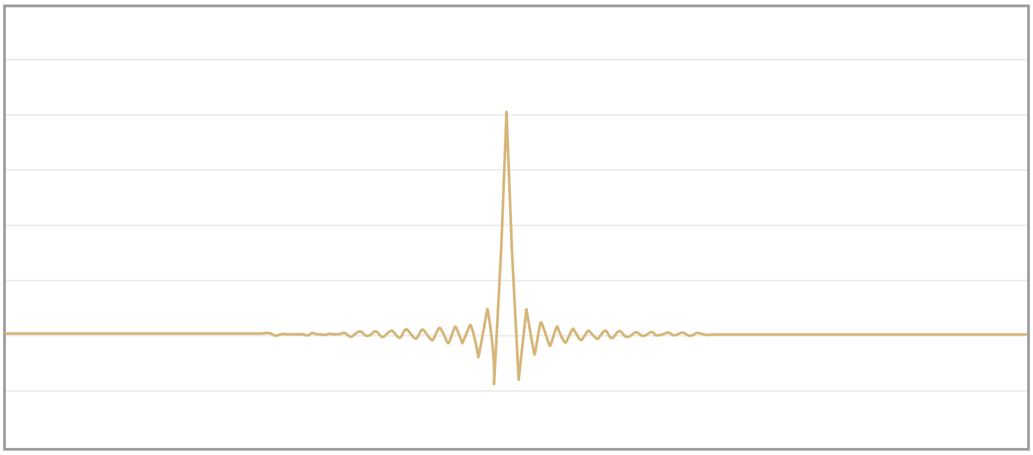
32 TAP FILTER FREQUENCY RESPONSE



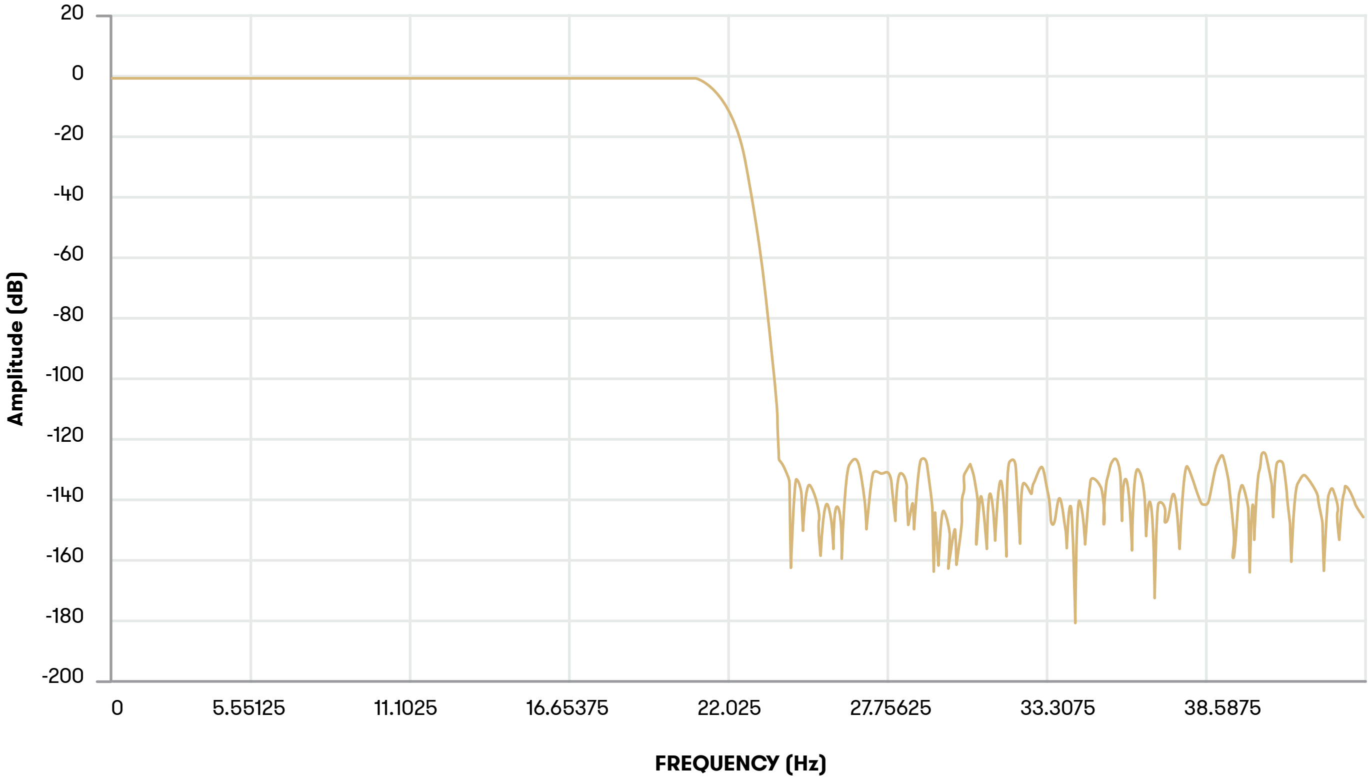
As can be seen from the above graphs, the frequency response is still far too droopy to be appropriate for use – at 20kHz, the response is still down about 2.8dB. Attenuation for images above 33kHz (or for images in the region of 0-11kHz) are 50dB down. The transition band width is in the region of 20kHz.

Next, the same Nyquist filter with a raised cosine window will be used, but this time with a length of 256 taps:

256 TAP FILTER IMPULSE RESPONSE



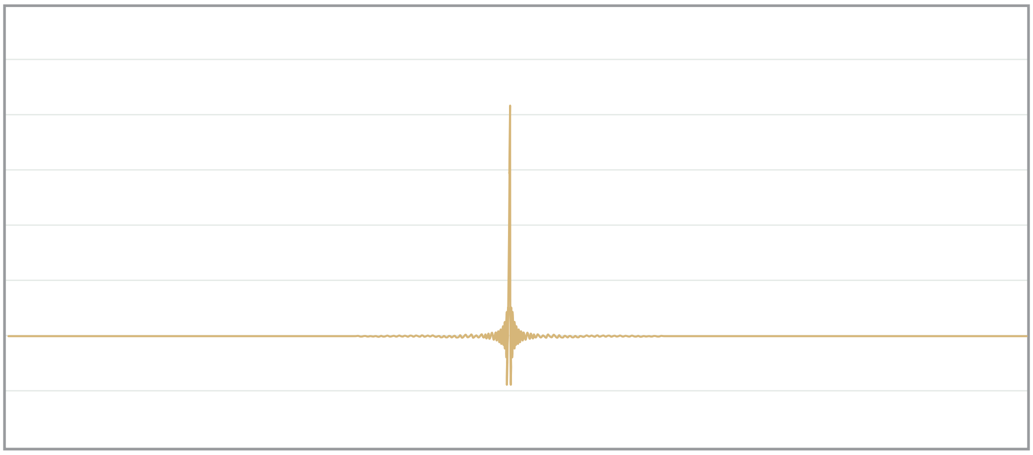
256 TAP FILTER FREQUENCY RESPONSE



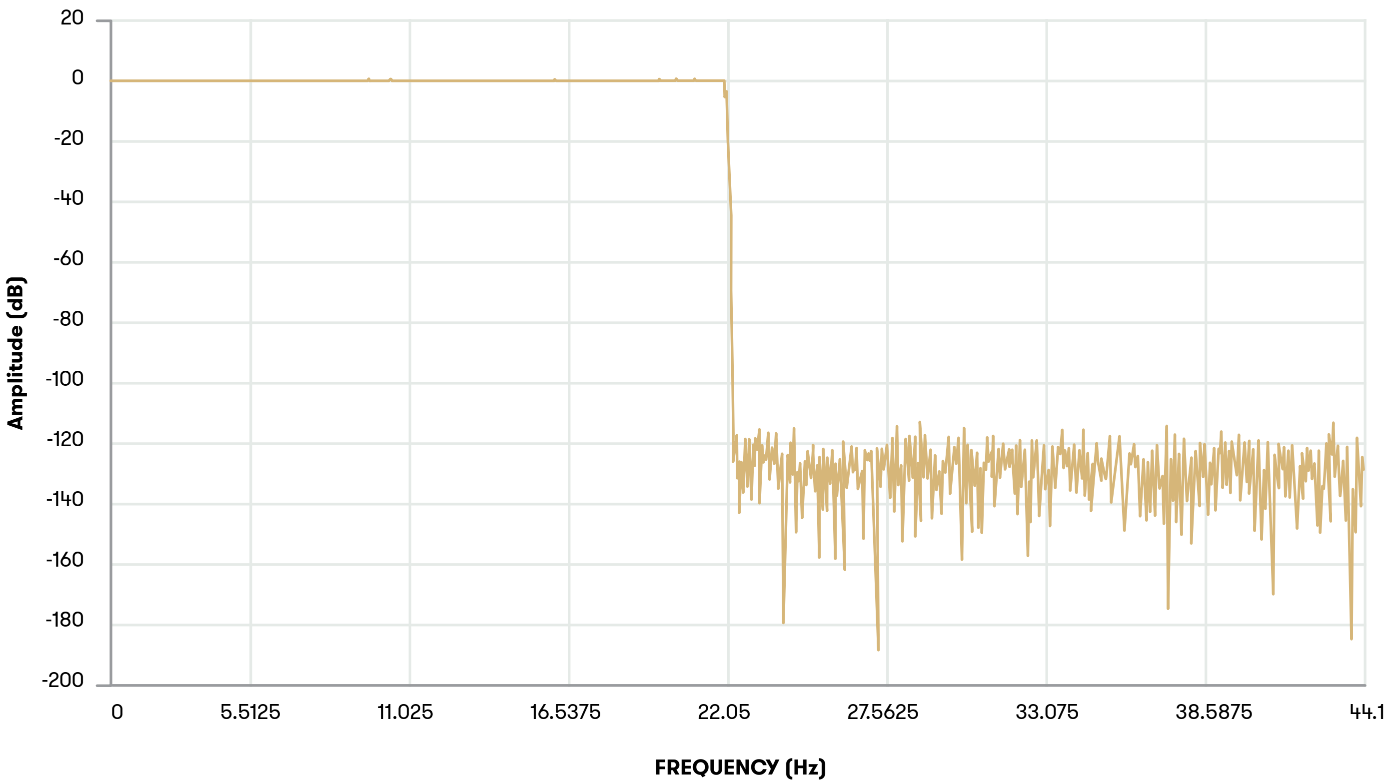
This response is visibly much better. The pass-band response is very flat up to 20kHz, and the transition band is approximately 4kHz wide. Images above 22.05kHz are supressed nicely. Looking at the impulse response however, it can be noted that much more data from *before* the current sample is playing is required. As such, a larger number of samples from 'the future' are required for this filter and will subsequently be having an impact on the current sample, and more samples from 'the present' will have an impact later on.

Lastly, using the same example upped to 1024 taps:

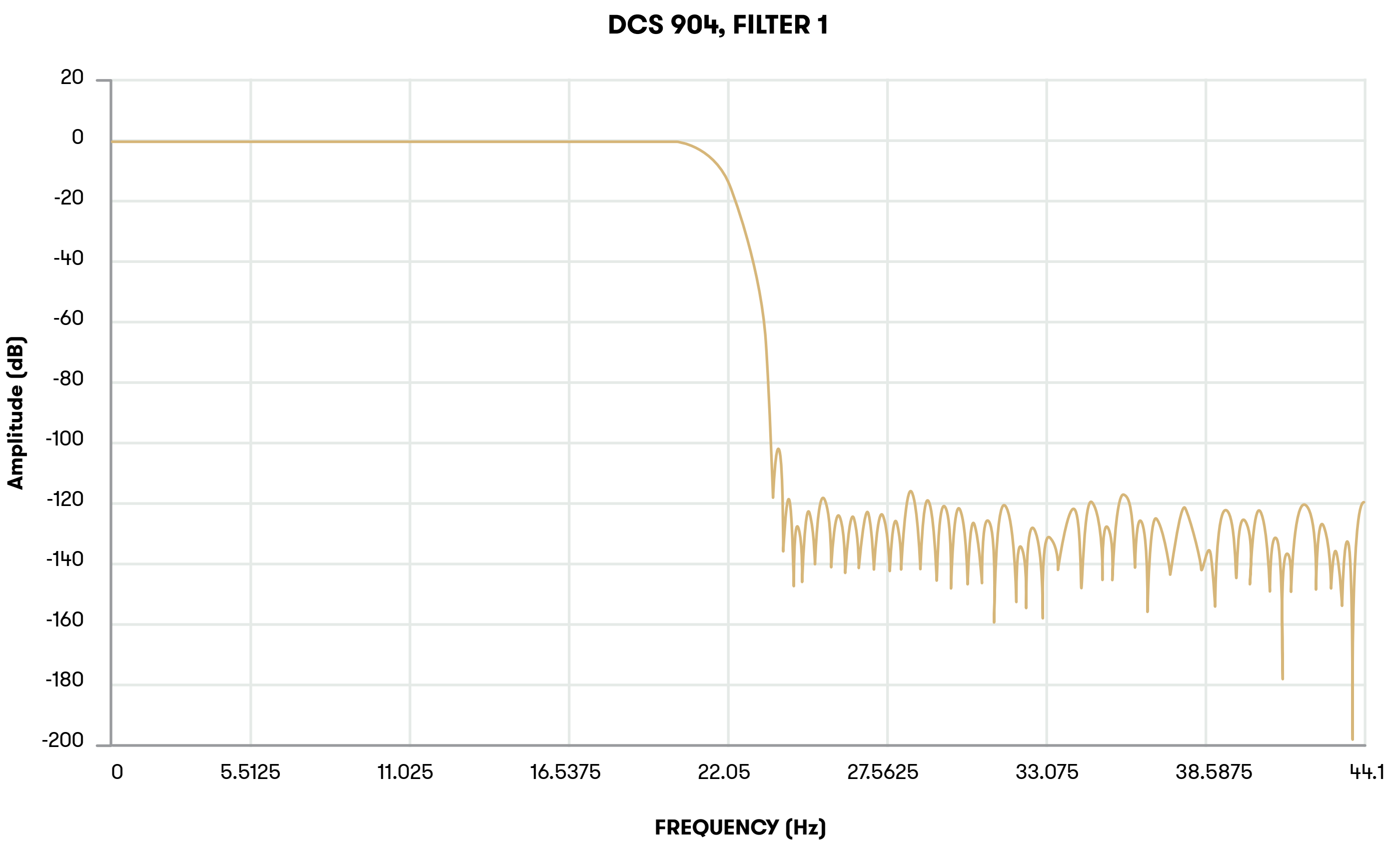
1024 TAP FILTER IMPULSE RESPONSE



1024 TAP FILTER FREQUENCY RESPONSE

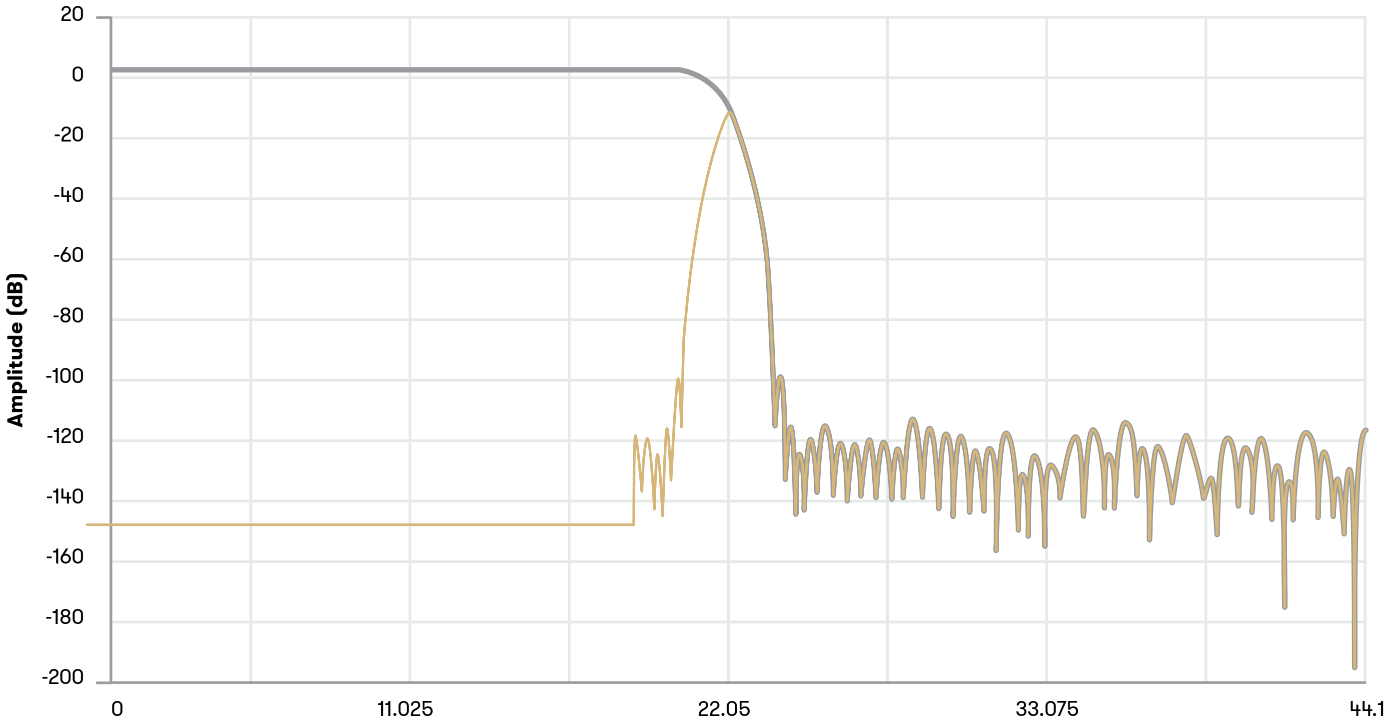


As can be seen here, lengthening the filter has a few effects. Firstly, the transition band has become much narrower, so there is less out of band energy. There are, however, some negative effects – because there are more coefficients performing more multiplies, the stopband noise rejection is actually starting to degrade. Noise is beginning to accumulate in the stop-band (which can be partly compensated for with the windowing). There are also now a large amount of samples from the 'future' being used (in this example of 2014 taps at 44.1k, around 11mS worth) and an equal amount will be affected in the 'past'. The effects of these factors are debatable, but **where** these samples come from has to be considered. With that in mind, here's a good real-world example…



This graph shows the frequency response of the dCS 904 running at 44.1kS/s using Filter 1.

This graph shows the filter frequency response from a very capable ADC, used in many recordings – a dCS 904. The first thing to note about this example is that it doesn't use a Nyquist filter like the examples above. The attenuation by the Nyquist frequency is 20dB. This is important as an ADC, by definition, deals with signals that are not bandlimited. The final filter is around 100 taps long, meaning that effectively, there is little to be gained from using a much longer filter on replay (inside the DAC). Considering what this response means, let's look at the chart below:



This graph helps to illustrate the area of uncertainty caused by this filter. In the intersecting area, it isn't possible to differentiate between real signal and an alias. As such, use of excessively long filters here leads to heavy use of DSP to preproduce what is effectively the transition band of the ADC – signal which is undesirable and unknown.

What this stands to show is that, with digital filter design, the signal chain as a whole needs to be considered, as opposed to just the DAC in isolation. DAC filters which are likely to work well with realistic ADC filters are ideal – in reality, the use of a filter in a DAC which is either not present, too short or too long can have detrimental effects.

Filter length must of course be balanced with the other factors described above, which is where good engineering comes into play – in other words, understanding how to employ the necessary trade-offs to create a set of filters which work well regardless of what content is put through it. This is the reason a dCS DAC has so many filters to choose from: the DAC doesn't (and can't) know the filters which were used to create the signal, so several options allow the user to achieve the best musical experience irrespective of source material.

"This is the reason a dCS DAC has so many filters to choose from: the DAC doesn't (and can't) know the filters which were used to create the signal, so several options allow the user to achieve the best musical experience irrespective of source material."

Our experience of designing both ADCs and DACs at dCS leaves us in a very strong position to be able to create DAC filters which consistently perform to the highest standards, both in testing and with real-world musical signals.

Filtering in digital audio

dcsaudio.com | **@dCSonlythemusic** | **info@dcsaudio.com**

Data Conversion Systems Ltd.
Unit 1, Buckingway Business Park, Anderson Road,
Swavesey, Cambridge, CB24 4AE, UK

dCS
ONLY THE MUSIC