

## FIR Filters Explained – Part 2

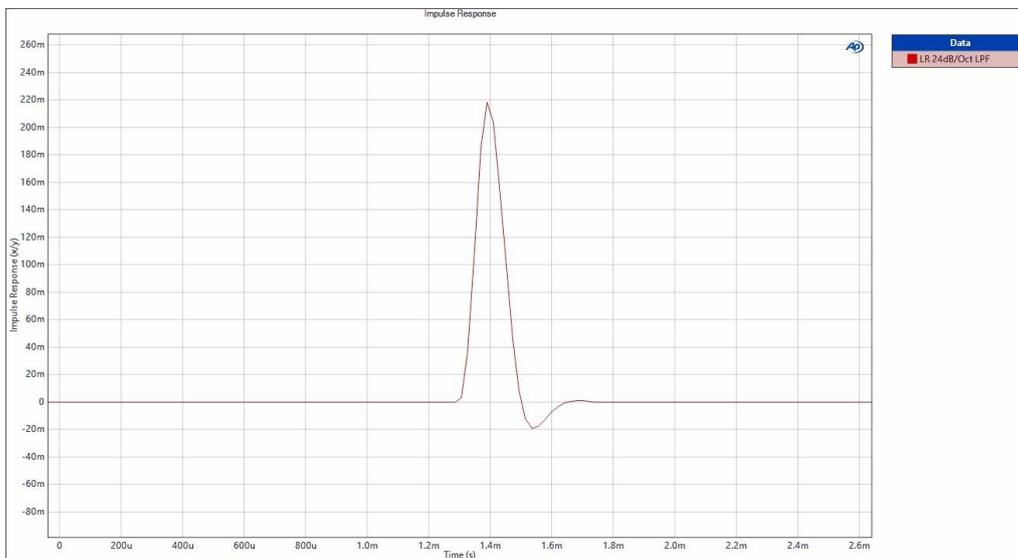
How they work & how they compare with IIR (standard) filters

### Introduction

FIR filters have become increasingly popular in pro-audio in recent years, driven by the availability of more powerful DSPs which are able to process multiple FIR filters in real-time. However, whilst FIR filters are a powerful tool they are also frequently misunderstood.

### Time Domain

Traditional audio engineering has focused on the frequency domain (so how the gain varies with frequency), especially obtaining a “flat” frequency response. In loudspeaker design and multi-way loudspeaker systems, we also need to consider the effect in the time domain due to filtering (so how the gain varies with time). By combining the frequency *and* phase responses of a system, we can mathematically apply an Inverse Fourier Transform to get the impulse response of a system. Don't worry if that doesn't mean anything to you – just know that the performance measurement data obtained from this maths is called the Impulse Response of the system and is vital to understanding how it responds in time and in frequency. This is the output of the system over time if you put a single impulse (an very short step ideally – think of a loud “click”) into the system. The impulse response of an IIR implementation of a Linkwitz Riley 24dB/octave low-pass filter would look like this:



The problem with the impulse response is that when we look at it, it's not very easy to understand what effect it will have on a signal. This is one reason why we tend to focus mainly on the frequency response and phase response, which are mathematically equivalent but provide more useful information. However, there are characteristics of the impulse response which have important implications, as we'll discuss later.

## **FIR Filters**

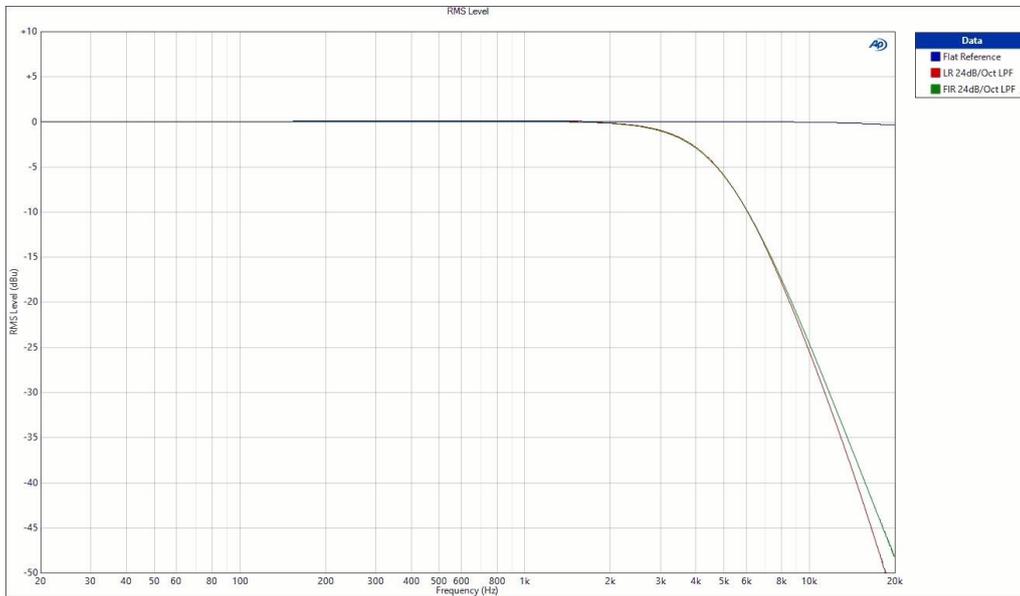
The alternative to IIR filters is the FIR filter which, as the name suggests, are designed using a fixed length impulse response. Actually, the normal design method for FIR filters is to design the frequency and phase response, then calculate an inverse Fourier transform to generate the impulse response (which is the FIR filter itself). From a design point of view, this has many advantages as it allows the design of filters with complex frequency responses without any of the side-effects of a non-linear phase response to cause problems in the acoustic system.

## **FIR Advantages**

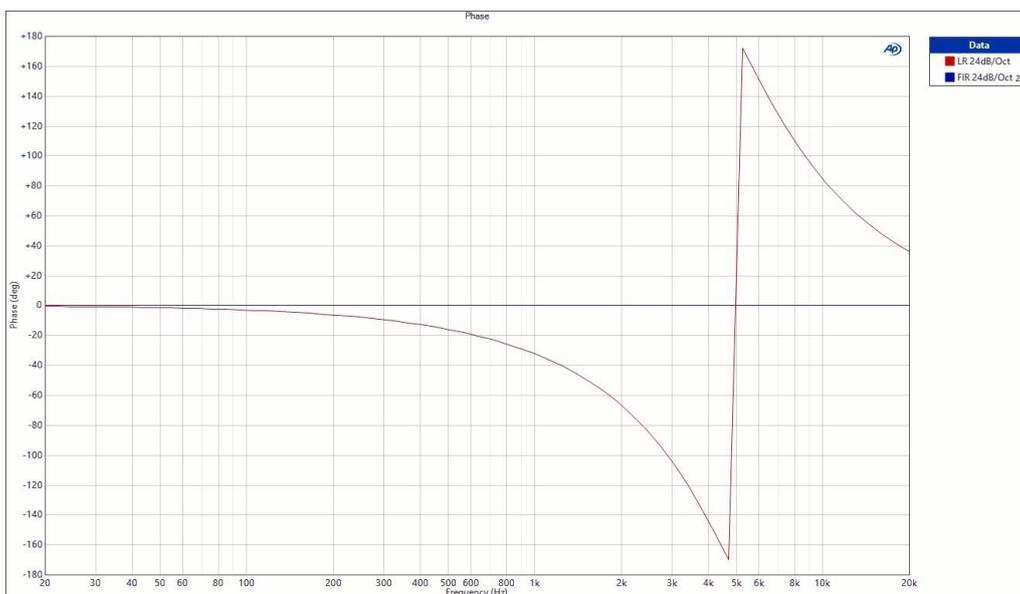
- Can create filters with arbitrary frequency response
- Have control over the phase response of the filter

## FIR Example – The Standard Crossover Again

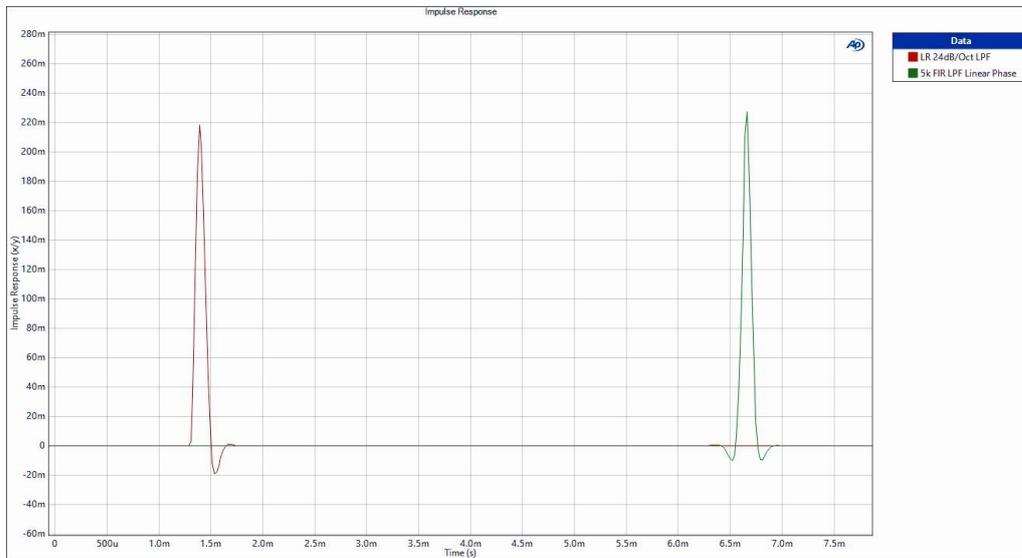
So we can design a 24dB/octave low-pass filter with (almost) exactly the same frequency response as our Linkwitz Riley filter in Part 1:



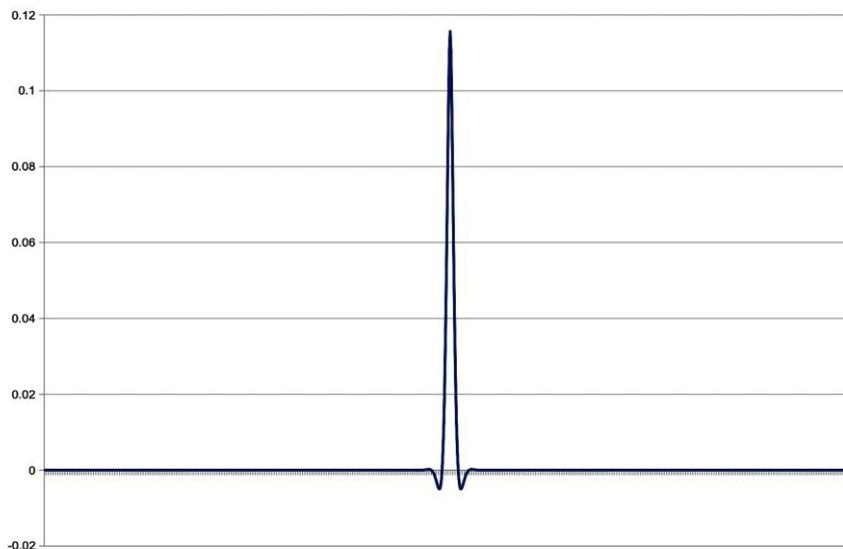
This filter has been measured using a 1024 tap FIR filter at 96kHz designed in rePhase, loaded into an NST D48S. This version of the filter has a completely flat phase response (compared against the IIR version for reference) :



However, we also need to look at the impulse response of our filter:



The measurements of both the FIR and IIR are shown here so we can see that it takes over 5ms longer for the impulse to come out of the FIR filter compared to our IIR filter with the same frequency response. This length of time is determined by the length of the FIR filter, or the number of "taps". In this case, the impulse response is exactly what we expect because it's the same as we designed in rePhase :



This long "group delay", the overall delay to the signal, is the price that we pay for a flat phase response. In fact it is the mechanism required to create the flat phase response. In some applications, a large group delay in the system isn't a problem but in real-time applications like live sound or gaming, it's a major consideration. The compromise is the accuracy of the filter versus the latency (time delay) it introduces – more accuracy means more "taps" so longer for a sample to pass through. This can be reduced by increasing the sample rate, so a 5mS latency at 48kHz becomes 2.5mS at 96kHz for the same number of taps.

There is one other feature of these impulse responses worth noting. The IIR filter impulse response is asymmetric, and is characterised by a single large peak (in response to the impulse input), followed by a decay which determines the behaviour of the filter. But for an FIR filter with linear phase response the impulse response is always symmetrical about the large peak (the position of which determines the group delay of the filter). This means that the filter starts generating audible output for several cycles before the main peak of the impulse has passed through the filter, creating a kind of “pre-ringing” on transients which in some circumstances can sound a little odd.

## **FIR Limitations**

So although we get some very desirable advantages using FIR filters, there are still disadvantages :

- Long group delay compared to IIR filters
- Require significantly more DSP power than a similar EQ/crossover in IIR filters
- Group delay length proportional to lowest frequency affected by the filter
- Only work over a limited frequency range for a given filter length
- Poor frequency resolution at lower frequencies

## **Conclusions**

Both IIR and FIR filtering techniques have applications in pro-audio, but care must be taken when choosing which is best suited for a particular application as both methods have benefits and drawbacks. There is sometimes a perception that FIR filters are superior, mainly because of the control they give over the phase response. While there are many powerful advantages to FIR filters, this functionality comes at a cost, not only in terms of group delay, but also the restriction of the frequency range which can be controlled.

## **Further Reading**

This article has focused on the concepts of FIR and IIR filters, rather than the engineering and mathematical details. The following links contain some more technical detail on FIR filters:

[An FIR tutorial on the Eclipse Audio site](#)

[Part 1 of a ProSoundWeb article on FIR Filters](#)

[Part 2 of a ProSoundWeb article on FIR Filters](#)