

Audio FIR Filtering

A Guide to Fundamental FIR Filter Concepts & Applications in Loudspeakers

Although not a new technology, manufacturers are increasingly including FIR (finite impulse response) filtering in loudspeaker processors and DSP based amplifiers due to the significant increase in performance-versus-cost of microprocessors and DSP hardware.

The advantages of FIR filtering include more arbitrary and fine control of a filter's magnitude and phase characteristics, independent control of magnitude and phase, and the opportunity for maximum-phase characteristics (at the expense of some bulk time delay). The primary disadvantage is efficiency; FIR filters are generally more CPU intensive than IIR (infinite impulse response) filters. For very long FIR filters, segmented frequency-domain and multi-rate methods help to reduce the computational load, but these methods come with increased algorithmic complexity.

In pro audio, the terms FIR Filter and FIR Filtering are often used when referring to specific implementations, such as:

- linear-phase crossover and linear-phase brick-wall crossover filters
- very long minimum-phase FIR based system EQ
- horn correction filtering

Whilst these implementations each have their uses, the capabilities of FIR filtering go beyond these implementations; particularly with regard to independent control of magnitude and phase, and mixed & maximum-phase characteristics.

So, what is FIR filtering and how does it compare to ubiquitous IIR filtering? This article aims to answer these questions, but doing so first requires covering a number of basic concepts in digital audio. If you've studied digital signal processing, much of this will be second nature. Forgive me for skipping some of the details and simplifying some of the more complex concepts.

1. Key Concepts in Digital Audio

Sampling

In digital audio, sound waveforms are represented by samples. An analog-to-digital converter (ADC) measures, or samples, an analog signal and assigns a digital value to each sample. Humans can typically hear frequencies between 20 Hz and 20 kHz. (A Hz is a cycle per second.) To adequately represent this frequency range digitally, the ADC needs to sample the audio waveform at least twice the highest audible frequency; hence we have the common sample rates of 44.1 kHz and 48 kHz. (Multiples of these frequencies, such as 88.2

kHz, 96 kHz, 192 kHz and 384 kHz are also used in pro audio; for reasons we won't go into here.) Half the sampling rate is called the **Nyquist frequency**. For example, a sampling rate of 48 kHz has a Nyquist frequency of 24 kHz.

For more information on the basics of digital audio and sampling, take a look at the "Monty" Montgomery's excellent [Digital Show and Tell](#) video and [24 192 Music Downloads](#) article on xiph.org.

Digital Filtering

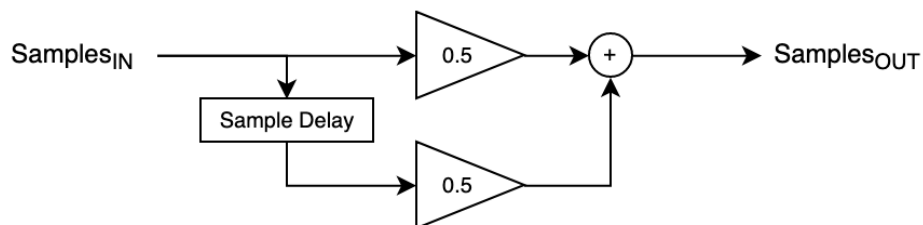
Digital filtering is a mathematical process for altering a digital audio signal. At each time interval – for a sampling rate of 48 kHz, the interval is 1/48000 seconds or 20.83 microseconds – a time-domain digital filter takes the current input sample and some previous input samples, scales (or multiplies) the samples by defined numbers, called filter coefficients, and sums the scaled samples to create an output sample. Let's look at some examples.

Example : "Averaging" Filter

One of the simplest digital filters involves taking the average of the current and previous samples. Conceptually this is:

$$\text{OutputSample} = (\text{InputSample} + \text{PreviousInputSample}) / 2$$

As a diagram, we can express the filter as:



Averaging digital filter flowchart.

As an equation, we can express this as:

$$y[n] = 0.5 * x[n] + 0.5 * x[n-1]$$

where

- $x[n]$ is the input sample at the current time interval, or sample number, n ,
- $x[n-1]$ is the input sample at the previous time interval, or sample number $n-1$,
- $y[n]$ is the output sample for the current time interval, or sample number n , and
- the "0.5" values are the filter coefficients.

Before continuing it's worth pausing to consider the **impulse response** and **frequency response** and how they relate to the filter structure above.

Impulse Response and Frequency Response

The impulse response of a device – an analog filter, a digital filter, a loudspeaker or even a room – is the reaction or response of the device over time in response to an input pulse.

Theoretically we can put a voltage, digital or acoustic pulse into a loudspeaker, amplifier, processor or room and record the output voltage, digital signal or microphone waveform over time to obtain the impulse response. However, a pulse doesn't have enough energy at all frequencies to excite a device with enough level to give a reasonably clean signal above ambient noise (or a signal-to-noise ratio; SNR) at all frequencies.

Today many measurement systems use either a sinusoidal sweep or cyclic pink noise. (Dual FFT methods use any audio signal – like the output of a live mixing console – but that's another story.) Sweeps and cyclic pink noise signals have enough energy at all frequencies to give reasonable SNR and therefore give a usable or stable impulse response.

The **frequency response** is the frequency-domain behaviour of the device. It's also the frequency-domain equivalent of the impulse response. (The term transfer function is sometimes used in place of the frequency response.) The frequency response is usually plotted as the magnitude (in dB) and phase (in degrees) across frequency. The frequency response can also be converted from magnitude and phase to a complex number representation; real and imaginary values for each frequency point.

Impulse Response \Leftrightarrow Frequency Response

The time-domain **impulse response** and frequency-domain **frequency response** are inherently linked and are mathematically-equivalent characterisations of the device. We can convert the impulse response to the frequency domain using the Discrete Fourier Transform (DFT) and convert the frequency response to the time domain using the Inverse Discrete Fourier Transform (IDFT); subject to some constraints that we won't go into here. The fast implementations of these operations are called the Fast Fourier Transform (FFT) and the Inverse Fast Fourier Transform (IFFT).

Cyclic pink noise

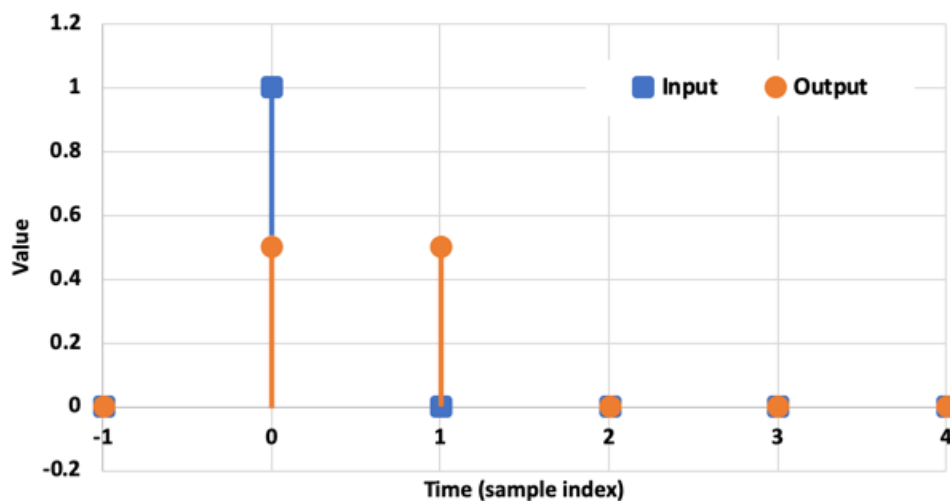
Cyclic pink noise is pink noise created, using FFT methods, in such a way that when the noise is played continuously by repeating the same noise sequence again and again, any captured chunk of the noise (of the same length as the original FFT sequence) is guaranteed to be pink. Cyclic noise makes measurement system calculations easier: a subject for another day. (In fact, cyclic noise of any spectral shape can be created using FFT methods. For clean, stable measurements it's often worth using noise with a spectral shape that matches the ambient noise spectrum so that all frequencies will have decent SNR.)

[... back to the "Averaging" Filter Example](#)

The impulse response of a digital filter is the output of the filter when we pass an impulse – a sample value of 1.0 – through the filter followed by zeros. Let's calculate the impulse response.

Sample number (Time Interval) n	Input Sample $x[n]$	Output Sample $y[n] = 0.5 * x[n] + 0.5 * x[n-1]$
-1	0.0	0.0
0	1.0	0.5
1	0.0	0.5
2	0.0	0.0
3	0.0	0.0
4, ...	0.0	0.0

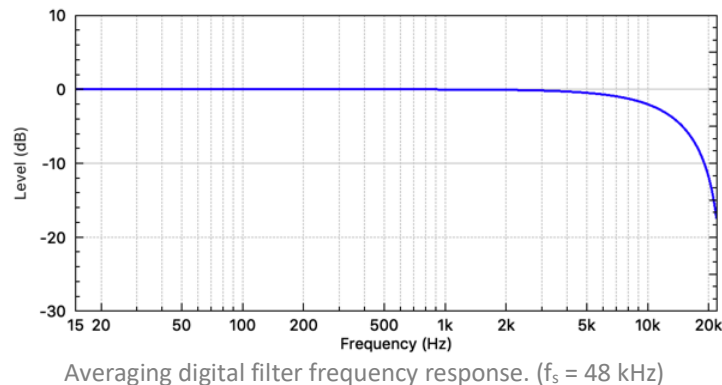
Averaging filter impulse response calculations.



Averaging digital filter impulse response.

For this filter, the impulse response is $[0.5 \ 0.5]$ for the 1st two time-intervals, and zero everywhere else. Its length is two samples, and since this length is finite, the filter is a **finite impulse response** or **FIR filter**.

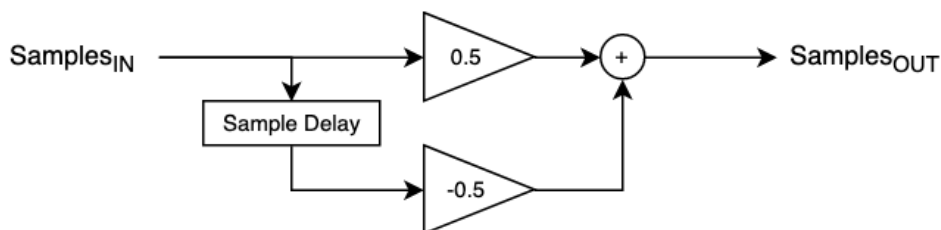
Since this filter “averages” pairs of samples, we would expect large sample differences to be smoothed out, and very low sample changes to mostly be unaffected. Let’s look at the frequency response magnitude.



Sure enough, the averaging filter is a **low-pass filter**. Low frequencies, where the audio signal varies more slowly over time, are unaffected, but high frequencies are attenuated.

Example : “Difference” Filter

Let’s look at what happens if we change the sign of one of the coefficients in the averaging filter. As a diagram, we can express the filter as:

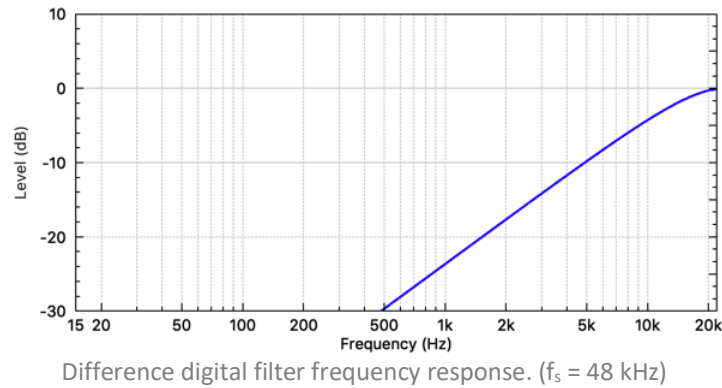


Difference digital filter flowchart.

As an equation, we can express this as:

$$y[n] = 0.5 * x[n] + -0.5 * x[n-1]$$

This filter cancels out adjacent samples that are the same, or similar, and emphasises pairs of samples that are very different. The difference filter’s frequency response is quite different to that of the averaging filter.



The difference filter is a **high-pass filter**. High frequencies are mostly unaffected, but low frequencies are attenuated.

Example : Digital Filter with Feedback

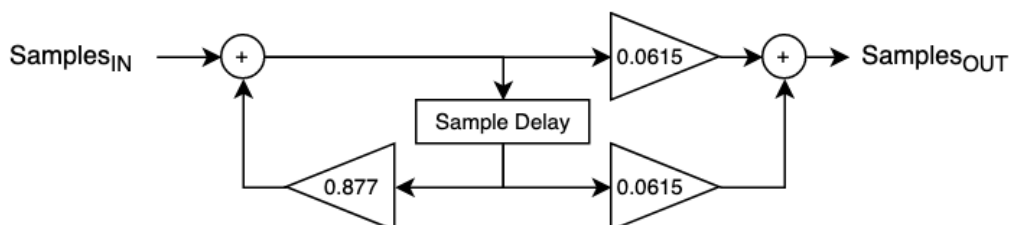
The filters in the two previous examples calculated the sum of a scaled history of samples. What if we take a previous or delayed sample, scale it, and feed it back into the sum?

The following diagram shows a Butterworth 1st order low-pass filter with a cut-off frequency of 1 kHz ($f_s = 48$ kHz). The important thing to note is that a portion of the previous sample is fed back, or re-circulated, into the filter.

Filter Order

For a digital filter, the filter order is the maximum amount of sample delay used in the digital filter.

For IIR low-pass and high-pass filters, the frequency response roll-off is 6 dB per octave multiplied by the order. (For example a 3rd order high-pass filter has an 18 dB/oct roll-off.)

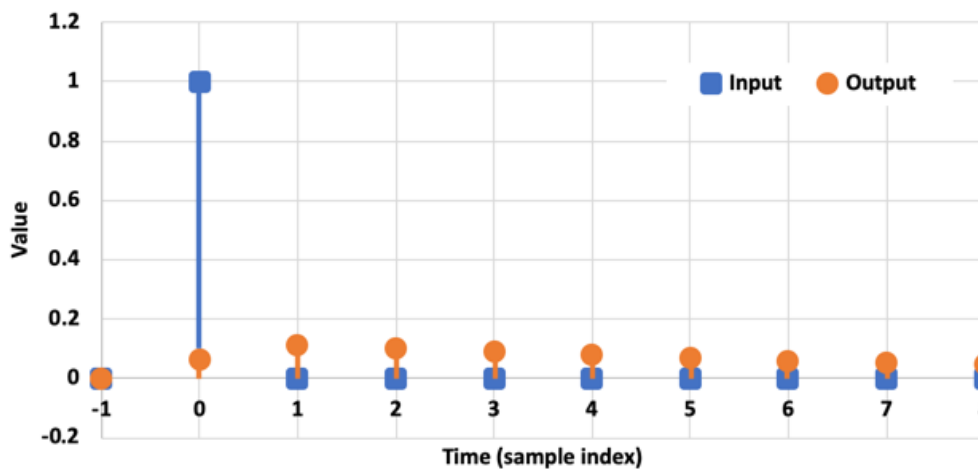


Butterworth 1st order 1 kHz low-pass filter flowchart. ($f_s = 48$ kHz)

Let's calculate the impulse response.

Sample number (Time Interval) n	Input Sample $x[n]$	Sample Delay Input $w[n] = x[n] + 0.877 * w[n-1]$	Output Sample $y[n] = 0.0615 * w[n] + 0.0615 * w[n-1]$
-1	0.0	0.0	0.0
0	1.0	1.0	0.0615
1	0.0	0.877	0.115
2	0.0	0.769	0.101
3	0.0	0.674	0.0888
4	0.0	0.591	0.0779
5	0.0	0.519	0.0683
6	0.0	0.455	0.0599
7	0.0	0.399	0.0525
8	0.0	0.350	0.0461
...
201	0.0	3.96×10^{-12}	5.21×10^{-13}

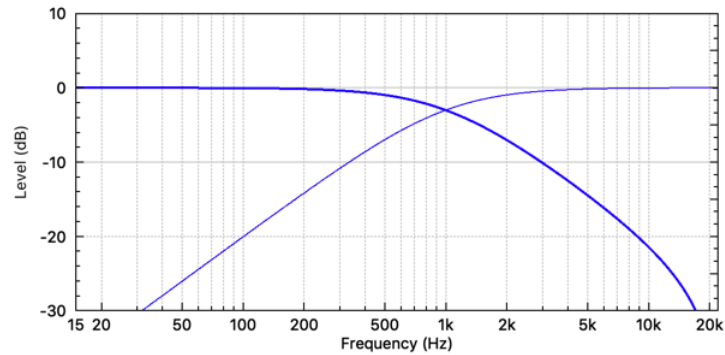
Butterworth 1st order 1 kHz low-pass filter impulse response calculations.



Butterworth 1st order 1 kHz low-pass filter impulse response. ($f_s = 48$ kHz)

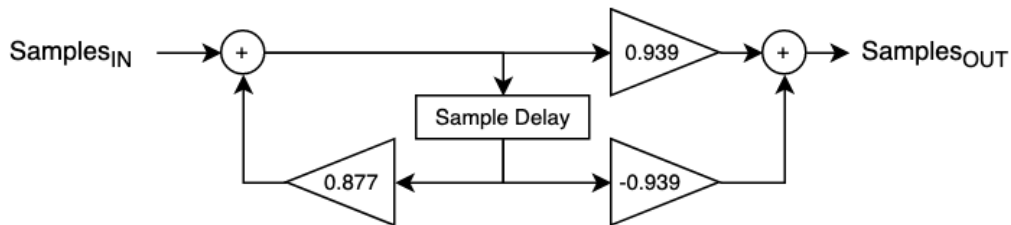
For this filter, the impulse response starts with a value of 0.0615 and, even though subsequent input samples are 0, the output of the filter decays but continues with non-zero values essentially forever. Since the output of the filter goes on for infinite time, the filter is called an **infinite impulse response** or **IIR filter**.

The thicker blue line in the following diagram shows the frequency response of the low-pass filter.



Butterworth 1st order 1 kHz low-pass and high-pass filter frequency responses. ($f_s = 48$ kHz)

The thinner blue line above shows the response of a 1st order high-pass Butterworth filter with a cut-off frequency of 1 kHz ($f_s = 48$ kHz). The following diagram shows the signal flow and coefficients for the high-pass filter.

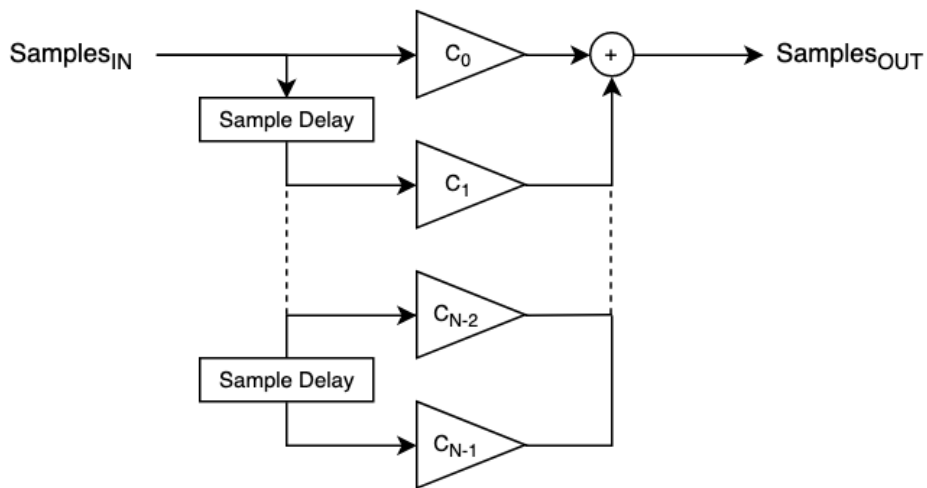


Butterworth 1st order 1 kHz high-pass filter flowchart. ($f_s = 48$ kHz)

In the following sections low-pass and high-pass filters will often be referred to as LP or HP filters.

2. The FIR filter

The following diagram shows the signal flow for a general FIR filter. The coefficients (C_0 to C_{N-1}) are the “taps” and so for a N length FIR filter there are N taps and $N-1$ sample delay elements. The filter order is $N-1$.

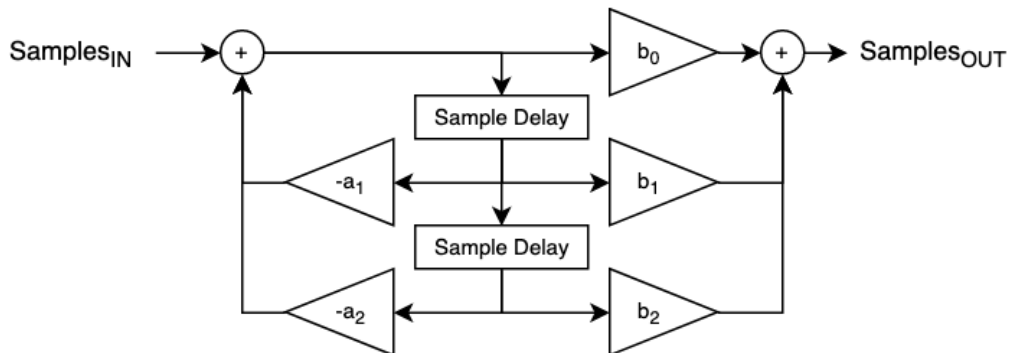


FIR filter flowchart.

While a short FIR filter can't do very much, longer FIR filters become quite powerful by essentially blending a long history of audio samples to cause level and phase changes at different frequencies in controlled ways.

3. The IIR Biquad

The second order IIR filter is typically called the biquad filter. (The following diagram shows the Direct Form 2 version.)



IIR biquad filter flowchart. (Direct-form 2)

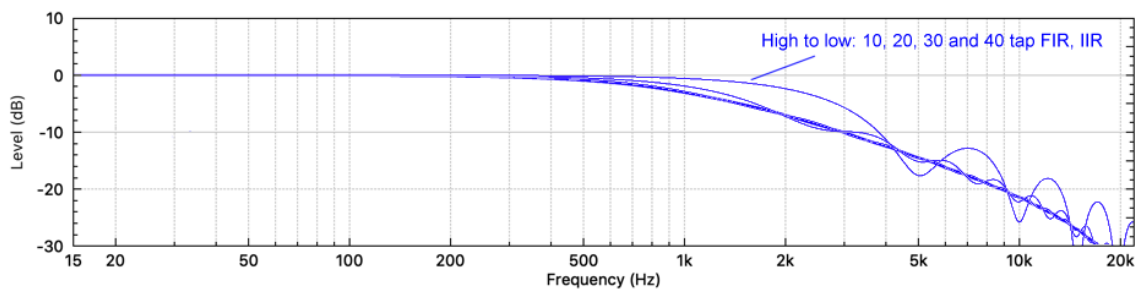
Biquads form the basis of most IIR filtering in DSP's. Shelf, parametric and all-pass filters can be implemented in this form, and high-pass and low-pass filters of any order are usually implemented as a cascade of connected biquad filters. A 1st order filter can be implemented in the biquad by setting coefficients a_2 and b_2 to 0.0. **Setting coefficients a_1 and a_2 to 0.0 turns the biquad into a 3-tap FIR filter.**

4. FIR versus IIR Filters

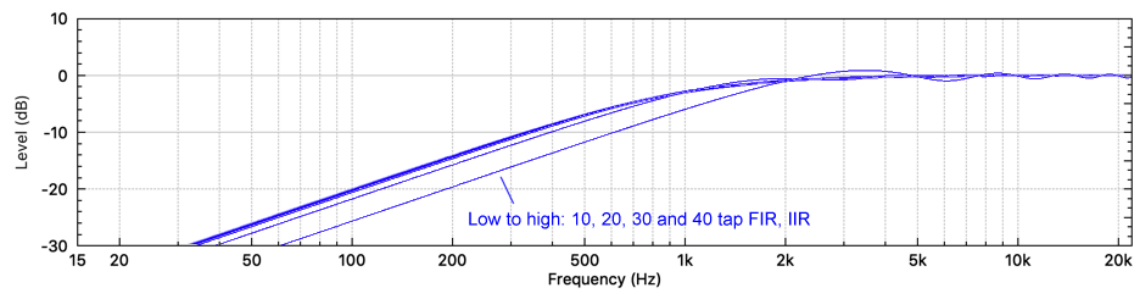
So how do IIR and FIR filters compare? Let's take a look at some common filter types.

Comparison : FIR filters vs 1st order IIR LP and HP filters

The following two plots show the frequency responses of 1st order Butterworth IIR LP and HP filters along with FIR filters of various lengths that are designed to approximate the IIR filters. (The FIR design method used involves sampling the IIR filter impulse response and applying DC correction.) Here the FIR filters need to be ~40 taps or longer to begin to accurately approximate the IIR filters. The 10, 20 and 30 tap FIR filters have significant ripple and deviate from the IIR, in magnitude, by up to ~6 dB.



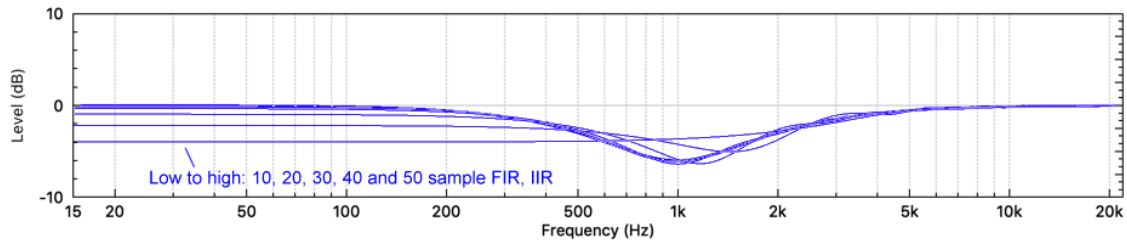
Comparing frequency responses of FIR filters with a Butterworth 1st 1 kHz low-pass filter frequency response. ($f_s = 48$ kHz)



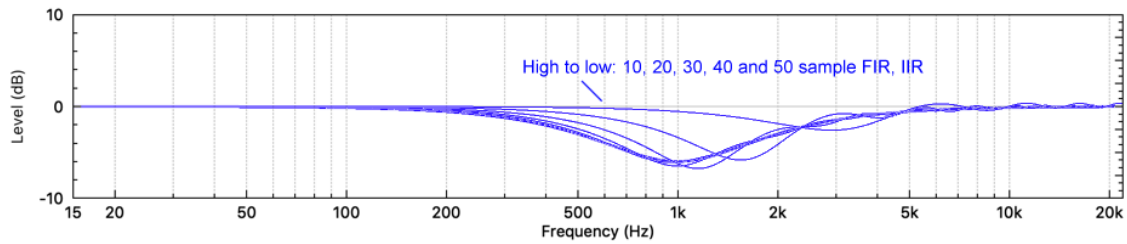
Comparing frequency responses of FIR filters with a Butterworth 1st order 1 kHz high-pass filter frequency response. ($f_s = 48$ kHz)

Comparison : FIR filters vs 2nd order IIR parametric filter

The following two plots show the frequency response of a 1 kHz Parametric IIR filter along with FIR filters of various lengths that are designed to approximate the IIR filter. Each plot shows a different FIR design method. The first method has more error toward DC but a slightly better match near 1 kHz. The second method matches the IIR better above and below 1 kHz but has a slightly worse match around 1 kHz. Using either method, the FIR filter needs to be ~40 taps or longer to begin to accurately approximate the IIR parametric filter.



Comparing Frequency Responses of 1 kHz Parametric IIR filter and FIR Filters from 10 to 50 Taps ($f_s = 48$ kHz)



Comparing Frequency Responses of 1 kHz Parametric IIR filter and FIR Filters from 10 to 50 Taps, with DC correction ($f_s = 48$ kHz)

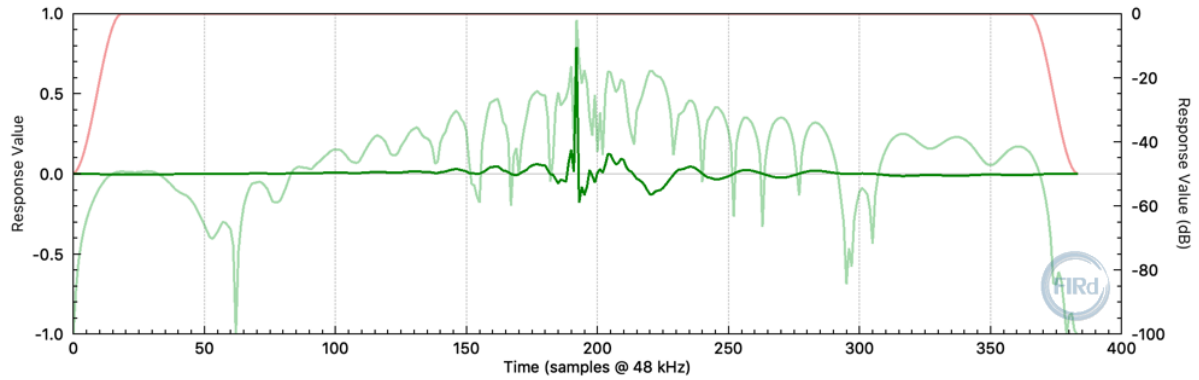
5. FIR Filter Length

Since FIR filters don't have feedback, their ability to affect low frequencies is directly proportional to their length. The longer the filter, the lower the frequencies that can be adjusted; either in magnitude, phase or both. Higher Q adjustments – sharper magnitude and phase transitions – also require longer FIR filters.

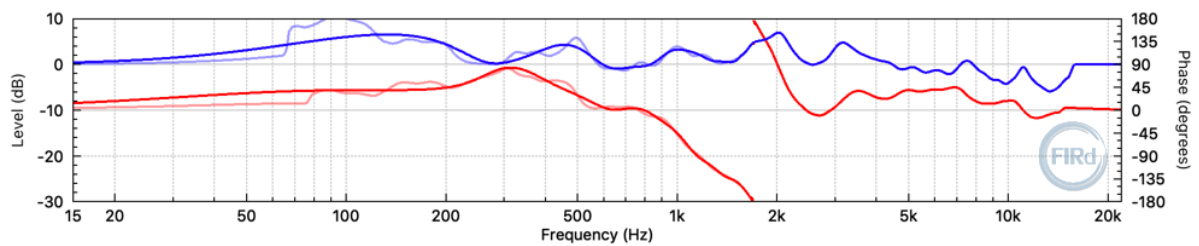
Following are examples of 384 and 3072 tap FIR filters; the filter responses are the dark-blue and dark-red lines. Both FIR filters are attempting to match the desired EQ for a loudspeaker – the light-blue and light red-lines. The difference plots show the difference in magnitude and phase between the desired ideal filter frequency response and the frequency responses of the FIR filters. Note:

- The longer the filter, the more effective FIR filtering is at achieving EQ, particularly toward low frequencies.
- Even the 3072 tap FIR filter can't achieve the high-Q magnitude change desired at ~65 Hz. (It actually takes over 10000 taps at 48 kHz for the FIR filter to match the desired EQ.)

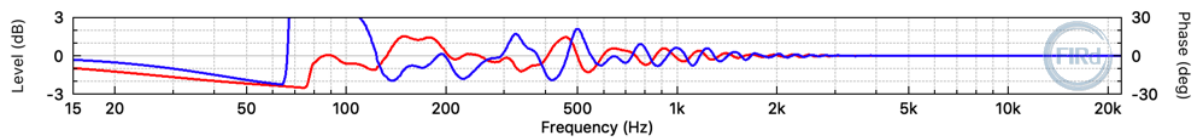
384 Tap FIR Filter



384 tap FIR filter impulse response (dark green) and dB magnitude of the impulse response (light green).

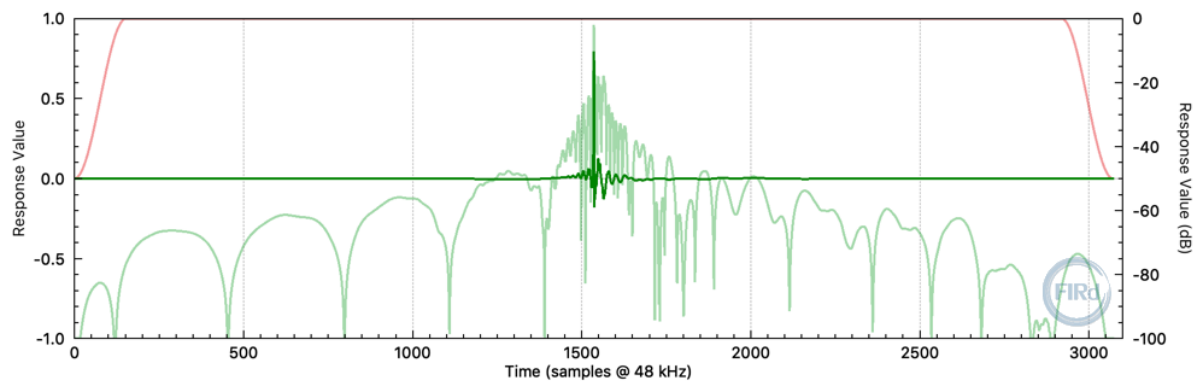


384 tap FIR filter frequency response (blue and red) and the desired ideal filter frequency response (light-blue and light-red).

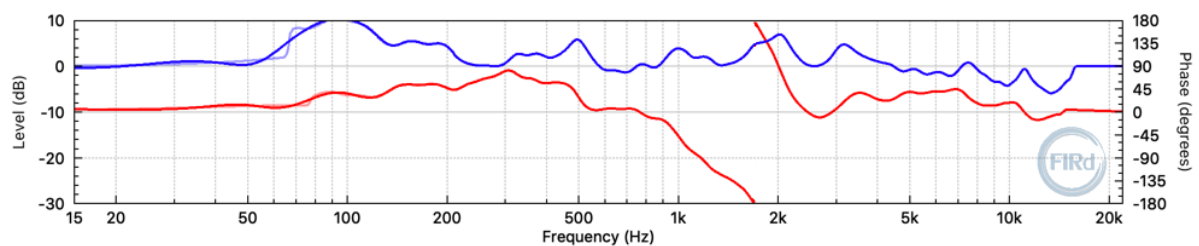


Frequency response difference between the desired ideal filter and the 384 tap FIR filter.

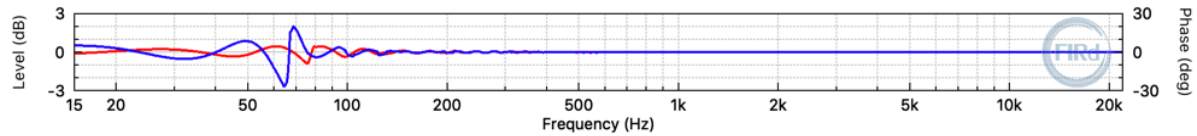
3072 Tap FIR Filter



3072 tap FIR filter impulse response (dark green) and dB magnitude of the impulse response (light green).



3072 tap FIR filter frequency response (blue and red) and the desired ideal filter frequency response (light-blue and light-red).



Frequency response difference between the desired ideal filter and the 3072 tap FIR filter.

6. Computational Complexity

In the introduction we mentioned that FIR filters are more computationally costly than IIR filters. Let's consider some of the simple IIR and FIR filters shown above. When estimating and comparing computational costs, we generally look at the mathematical operations – multiplications and additions. We assume that a processor can calculate a “multiply” and an “add” effectively in the same operation; and therefore we can ignore the additions and just count and compare the multiplications.

The 1st order IIR filter above has 3 coefficients which need to be multiplied with the audio samples, and so we estimate the filter to take approximately 3 x the sample rate operations per second. The FIR filter has N coefficients (where N is the filter length) and so we estimate the FIR filter to take N x the sample rate operations per second.

The following table compares the filters from above.

fs = 48 kHz	IIR	FIR (N=10)	FIR (N=20)	FIR (N=30)	FIR (N=40)
1st Order LP or HP Filter	144k	480k (3.3x)	960k (3.3x)	1,440k (10x)	1,920k (13.3x)
2nd Order Parametric Filter	240k	480k (2x)	960k (3.3x)	1,440k (6x)	1,920k (8x)

Comparing multiplications per second for IIR and short FIR filters. (All the 'x' numbers are the FIR multiplications divided by the multiplications the IIR filter in the same row.)

We've seen in the examples how approximately 40 taps or more are needed for the FIR filters to approximate the IIR filters, and the table shows that this comes at the computational cost of 8 times or 13.3 times that of the IIR.

Typical Speaker Processor output Channel

As at the writing of this article, a common high-end speaker processor has approximately 24 IIR biquads for high-pass, low-pass, shelf and parametric filters, and 2048 taps of FIR. The following table compares the computational cost of both.

fs = 48 kHz	Multiplies
24 IIR Biquads for HP, LP, Shelf and Parametric Filters	0.576M
2048 Tap FIR Filter	9.83M (17x)

Comparing multiplications per second for IIR and FIR filters for a typical DSP output channel. (The 'x' number is the FIR multiplications divided by the multiplications of the IIR filters.)

7. FIR Filter Benefits

If FIR filtering is so computationally costly, what are its advantages? There are two primary benefits:

- independent control of magnitude and phase, and
- more detailed equalisation (including easier filter creation from a desired frequency response).

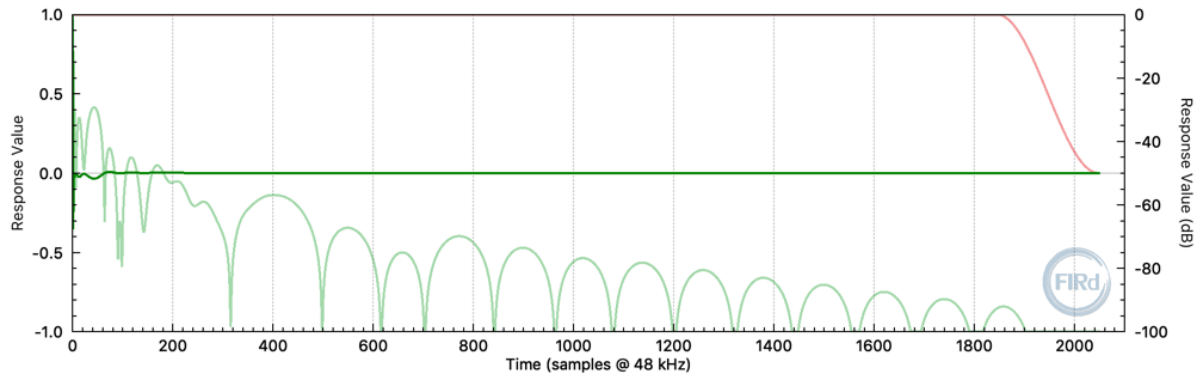
Let's explore each of these further.

Independent Control of Magnitude and Phase

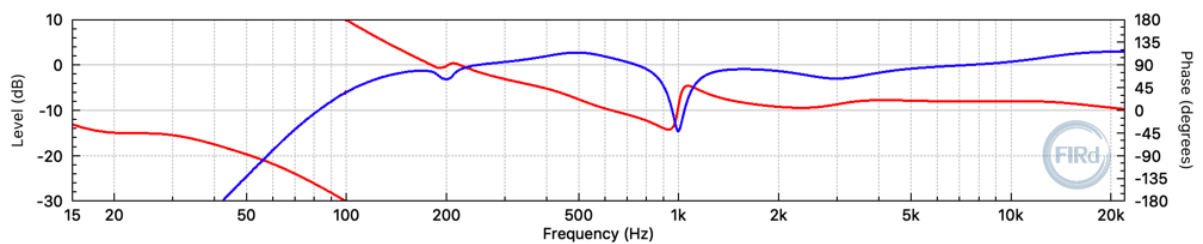
With most IIR filters, the phase response is inherently linked with the magnitude response. (One exception is the IIR all-pass filter.) A huge benefit of FIR filtering is the ability to manipulate magnitude and phase independently. Following are four FIR filter examples. Each have the same magnitude response but very different phase responses.

Example : Minimum-phase FIR Filter

Previously we showed how both IIR and FIR filters use sample delays (as well as coefficients) to achieve their intended changes in the frequency response. A minimum-phase filter effects EQ whilst adding the least amount of delay to the audio signal. (This is one of the reasons long FIR filter based EQ in PA systems is typically minimum-phase.) A characteristic of a minimum-phase filter is that its impulse response has larger coefficients at or near the start of the impulse response. The following two plots shows a minimum phase FIR filter that effects a HP near 100 Hz and some EQ. Whilst the FIR filter length is 42.7 ms, the effective delay is negligible.



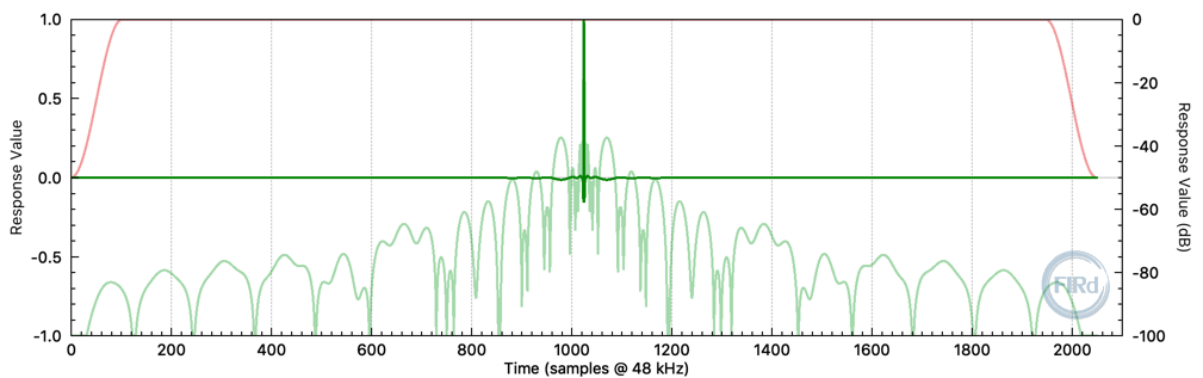
Minimum-phase 2048 tap FIR filter impulse response (dark green) and dB magnitude of the impulse response (light green). ($f_s = 48$ kHz)



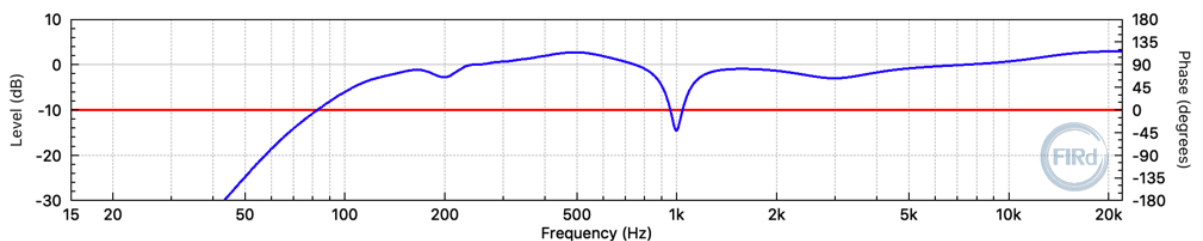
Minimum-phase 2048 tap FIR filter frequency response. ($f_s = 48$ kHz)

Example : Linear-phase FIR Filter

The following two plots show a FIR filter with the same magnitude response but with a flat or linear phase. The bulk delay through the filter is equivalent to the peak location of the filter: here 1024 samples or 21.3 ms.



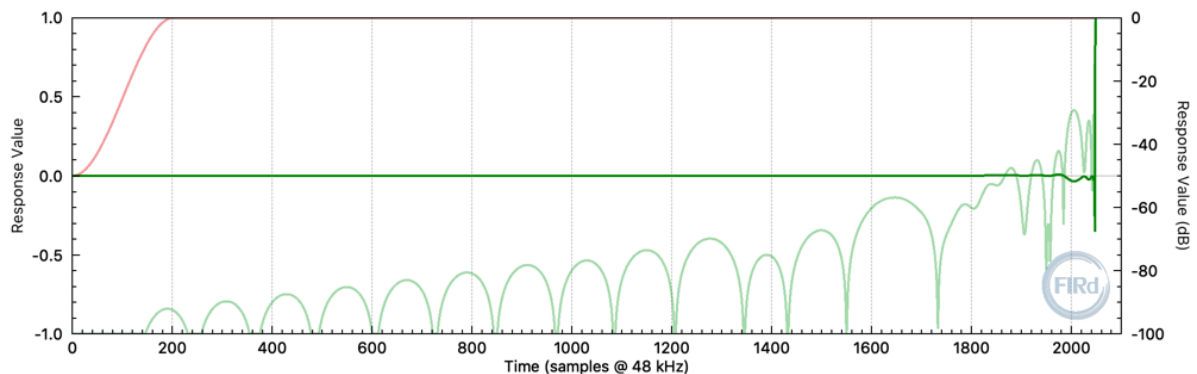
Linear-phase 2048 tap FIR filter impulse response (dark green) and dB magnitude of the impulse response (light green). ($f_s = 48$ kHz)



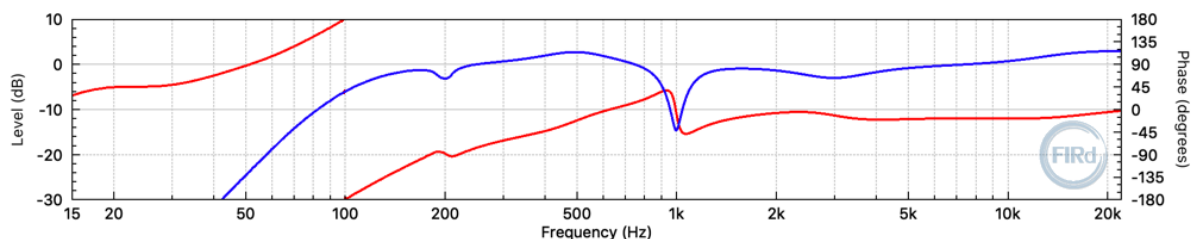
Linear-phase 2048 tap FIR filter frequency response. ($f_s = 48$ kHz)

Example : Maximum-phase FIR Filter

The following two plots show a FIR filter with the same magnitude response but with maximum-phase; this is the opposite or inverse phase of the minimum-phase filter above. The impulse response is the time reverse of the minimum-phase impulse response and so the bulk delay through the filter is approximately the length of the filter; 42.7 ms.



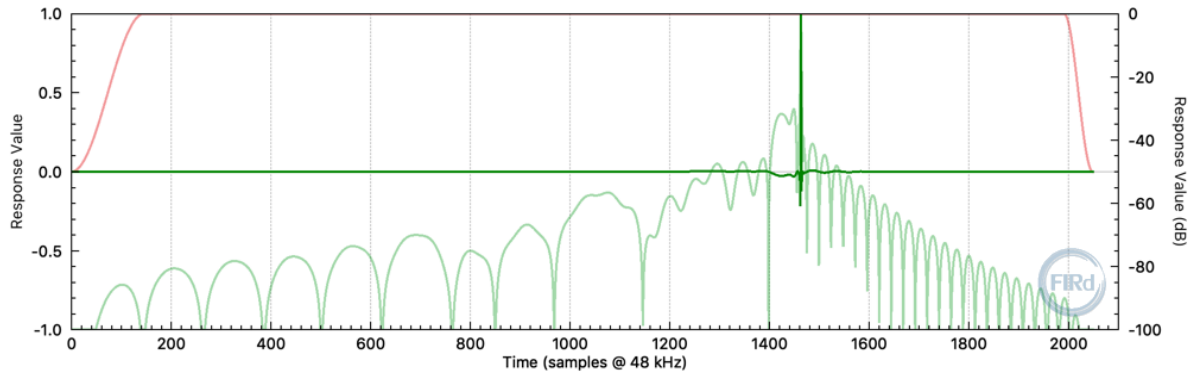
Maximum-phase 2048 tap FIR filter impulse response (dark green) and dB magnitude of the impulse response (light green). ($f_s = 48$ kHz)



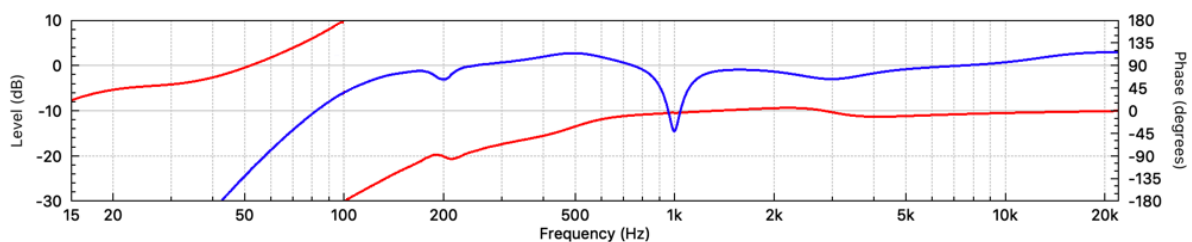
Maximum-phase 2048 tap FIR filter frequency response. ($f_s = 48$ kHz)

Example : Mixed-phase FIR Filter

Finally we have an arbitrary phase or mixed-phase FIR filter with the same frequency response. The bulk delay through the filter is approximately the location of the filter peak; here ~1480 samples or 30.4 ms. Where the peak is placed depends on the desired characteristics of the FIR filter and how those characteristics can be achieved within the tap length limit; here 2048 taps. To better understand this, take a look at some of the [FIR Designer tutorials](#).



Mixed-phase 2048 tap FIR filter impulse response (dark green) and dB magnitude of the impulse response (light green). ($f_s = 48$ kHz)



Mixed-phase 2048 tap FIR filter frequency response. ($f_s = 48$ kHz)

Why do we care about mixed-phase behaviour? So we can push a loudspeaker's phase to where we want it!

Why is Independent Phase Control Useful?

A loudspeaker driver can be thought of as a minimum-phase filter (when comparing the acoustic output with the electrical signal into the driver). When using minimum-phase EQ to bring a loudspeaker driver's magnitude response closer to "flat," the phase of the loudspeaker driver also flattens and moves closer to linear phase (at least within the audible pass-band of the loudspeaker).

However, in a typical multi-way loudspeaker the IIR HP and LP crossover filters (as well as polarity, delay and acoustic filters, like ports) all add frequency-varying extra phase. Because of this extra phase, a multi-way loudspeaker can be thought of as a minimum phase system PLUS some all-pass filters.

Since minimum-phase EQ generally doesn't affect the all-pass behaviour, we can use FIR filtering to move the phase of the loudspeaker to where we want it.

Arbitrary phase manipulation has many applications including;

- Phase linearising a loudspeaker. (Despite the apparent improvement in the loudspeaker impulse response, there's some debate as to whether this gives a perceptual improvement in loudspeaker performance.)

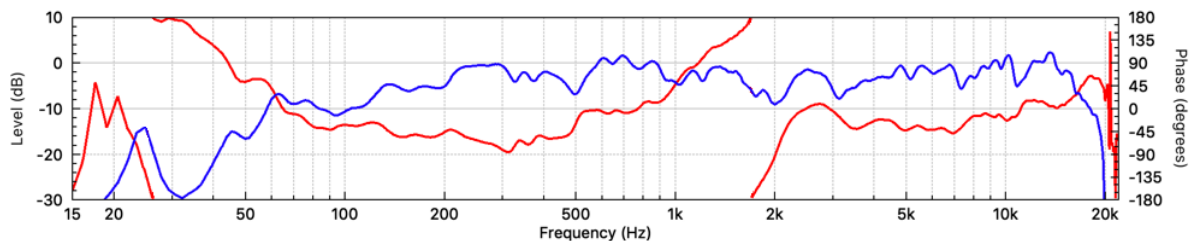
- Matching the phase (and magnitude) of loudspeakers within product lines, and across different models in installs, so that they are easier to tune in clusters and to array together.
- Manipulating individual loudspeakers in array processing (for audience coverage optimisation) and in beam steering.
- Crossover optimisation to improve frequency response consistency within the coverage angle of a multi-way loudspeaker.

More Detailed Equalisation (& Easier Filter Creation)

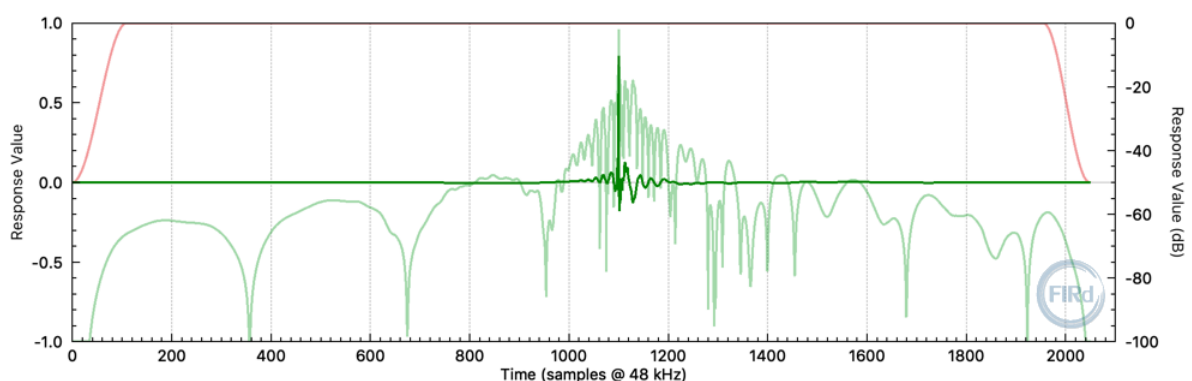
Using a loudspeaker measurement, we can create a frequency response (magnitude and phase) that will push the loudspeaker towards a desired target response. Because of the inherent relationship between the impulse response and frequency response, FIR filter coefficients can be generated from a desired frequency response fairly easily using DFT (or FFT) methods. The target response can be anything including:

- pink noise flat
- pink noise flat with slight HF roll-off (such as Cinema X-Curve)
- flat (linear) phase
- the magnitude and phase response of another loudspeaker
- the magnitude and phase response required from array processing calculations

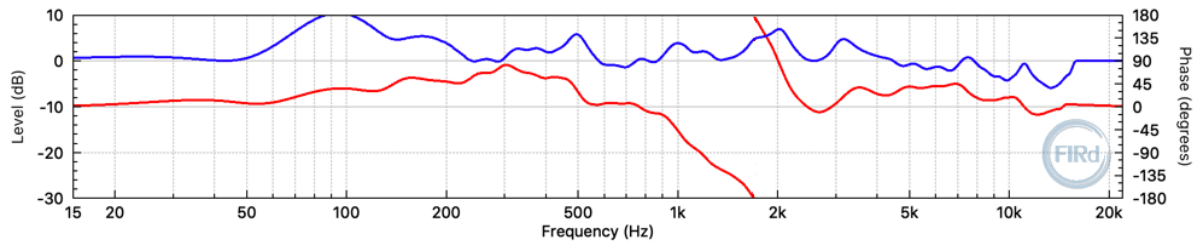
The following four plots show an on-axis measurement of a commercial 12" + horn 2-way cabinet and a 2048 tap FIR filter created to push the cabinet's response to have a relatively flat magnitude response (with a slight HF roll-off) and flat phase in the pass-band.



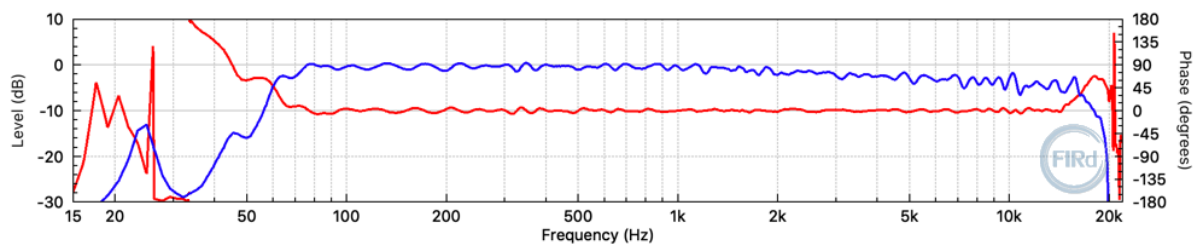
On-axis frequency response of a 12" + horn 2-way cabinet. (Time=0 is aligned to the HF driver and so the LF driver is relatively forward in time, resulting in the apparent maximum-phase behaviour at the crossover frequency.)



Impulse response (dark green) of FIR filter created for the 12" + horn 2-way cabinet, and dB magnitude of the impulse response (light green).



Frequency response of FIR filter created for the 12" + horn 2-way cabinet.



Frequency response of the FIR filtered 12" + horn 2-way cabinet.

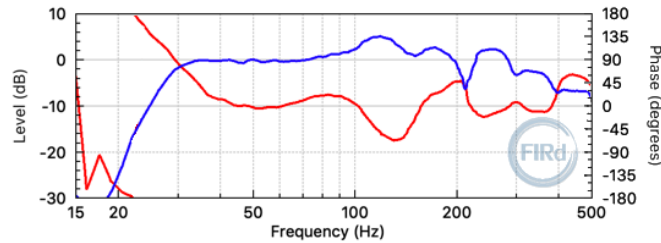
Designing EQ from Measurements and Measurement Averaging

Attempting to EQ fine structure and ripples in loudspeaker responses can be problematic. A loudspeaker's response varies with microphone and loudspeaker position, with level & temperature, and even over time. Fine EQ designed from a single measurement might result in the loudspeaker sounding better for the conditions and position of the measurement, but is likely to make the loudspeaker sound worse at other positions and at other levels etc. Great care must be taken to ensure the measurement is relevant and useful under all the conditions the loudspeaker will be used in – for example within the whole of the intended coverage area. Averaging measurements from multiple locations is one way to create a measurement that can be used as a starting point for effective, fine EQ.

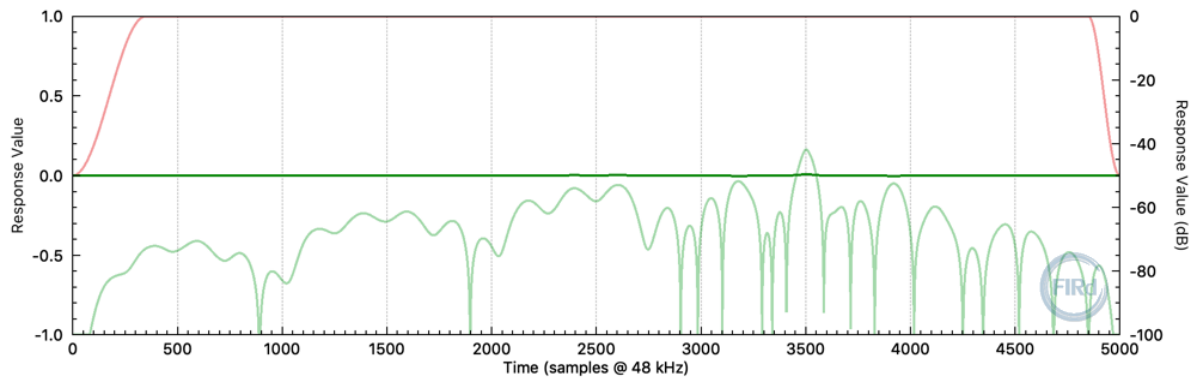
8. FIR Filtering for Subwoofers?

The following plots show an example of using a FIR filter to both EQ a subwoofer and unwrap the low-frequency phase (from the cabinet and the high-pass IIR used to protect against over-displacement).

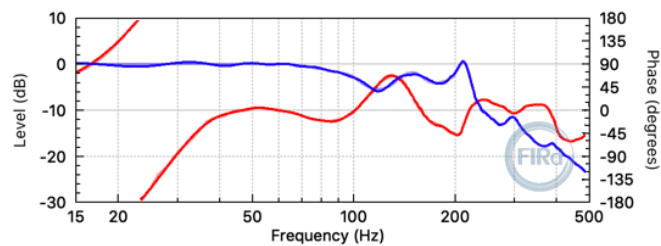
Filtering at very low frequencies requires very long filters; here 5000 taps with an IR peak delay of 3500 taps or 72.9ms. (This is simply an example of what a FIR filter can do and it's yet to be tested whether the reduction in low-frequency group delay improves the perceived sub impact. With such a large delay, it's probably not useful for live applications but may be useful in cinema and home theatre applications.)



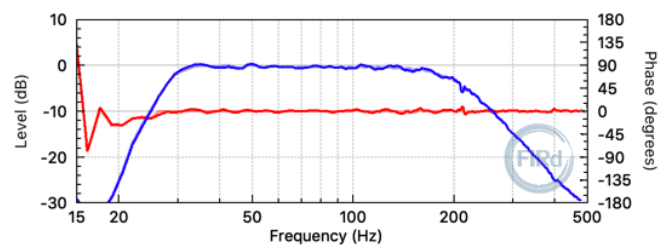
Subwoofer frequency response, before FIR filtering. (Double 18".
Measurement includes ~30 Hz 18dB/oct Butterworth IIR high-pass.)



Impulse response (dark green) of FIR filter for EQ, phase unwrapping and crossover LPF. (5000 taps and 3500 sample delay.)



Frequency response of FIR filter for EQ, phase unwrapping and crossover LPF. (5000 taps and 3500 sample delay.)



Subwoofer frequency response, with FIR filtering.

9. Designing & Loading Custom FIR Filters

The growing awareness of the benefits and flexibility of FIR filtering in audio, combined with the ever-increasing performance-versus-cost of microprocessors and DSPs, has resulted in increasing numbers of audio products with user-accessible FIR filtering blocks. These

products enable loudspeaker designers, installers, system operators and DIY'ers to load custom FIR filters. See a list of [FIR-capable processors, amplifiers & software products](#).

Software tools like [FIR Designer](#)* enable the design and simulation of FIR based EQ and mixed IIR+FIR presets/tunings for loudspeakers and systems, from loudspeaker or system measurements. Comprehensive measurement averaging functionality for spatial and level averaging is also included.

Written by Michael John, founder of [Eclipse Audio](#).

* All the FIR filter examples and plots in this document were generated with [FIR Designer](#).