

Sound Design Stack Exchange is a question and answer site for sound engineers, producers, editors, and enthusiasts. It only takes a minute to sign up.



Sign up to join this community

Anybody can ask a question

Anybody can answer

The best answers are voted up and rise to the top

Sound Design

What does 'Poles' mean in relation to a filter?

Asked 9 years, 9 months ago Active 1 year, 7 months ago Viewed 48k times



28

Sometimes filters are referred to in terms of dB/Octave (a 24dB/O is a common low pass filter setup, for example), however sometimes they are referred to in terms of 'Poles' (I've seen switchable 2/4Pole LPFs).



22

What does 'Poles' refer to when you are talking about filters, and how does changing the number of 'Poles' change the resulting sound?



equalization filter

edited Apr 1 '14 at 18:17



mjcopples

168 4

asked Dec 24 '10 at 11:17

Ed J

See maximintegrated.com/app-notes/index.mvp/id/733 – endolith Jan 19 '13 at 6:09

3 Answers

Active Oldest Votes



It's just a way of saying how many dB per octave a filter has.

11 A simple filter has a slope of 6 dB per octave. A two-pole has a slope of 12 dB/oct, and 4-pole 24 dB/oct.



As a further explanation to this, some synths have two 2-pole filters that can be either high-pass or low-pass, so they can be configured as a 24dB/oct lowpass, 24dB/oct high pass or a 12dB/oct bandpass filter (and very unusually, a 12dB/oct notch filter).

answered Dec 25 '10 at 9:16



Lennart Regebro

1,420 6 9

6 This is not strictly true. While we tend to use #-poles to denote the slope of LP/HP/BP filters, number of poles and zeros does not necessarily correspond to the slope of these canonical designs. Filter design follows Butterworth's statement: "An ideal electrical filter should not only completely reject the unwanted frequencies but should also have uniform sensitivity for the wanted frequencies." But pole-zero analysis can be used to explore all sorts of filter designs such as comb filters. Indeed comb filters typically have many more than 4 poles. Wikipedia on combs: bit.ly/fkjgrz – msutherl Jan 1 '11 at 21:21

True, it's only relevant for LP/HP/BP filters (and filters also typically don't have a dB/oct anyway). –

Lennart Regebro Jan 1 '11 at 22:39

2 Furthermore, there are low-pass or high-pass filters which achieve steeper slopes than these. They do so at the expense of other parameters, such as ripple. The numbers you cite apply only to Butterworth filters. – **Phil Frost** Mar 7 '14 at 15:40



50



In the digital domain, "Poles" come from the Z-transform of the impulse response of a filter, known as the transfer function.

Impulse Response

The [impulse response](#) of a [filter](#) is the output when you feed it an impulse. An [impulse](#) is signal that looks like this: [1 0 0 0 0 ...] – a pure digital click. The impulse response is important because it turns out that it is sufficient to characterize any [linear, time-invariant \(LTI\)](#) filter completely. You can [convolve](#) a signal by the impulse response of a filter to 'apply' that filter to the signal. This is how [convolution reverb](#) works, for instance: an impulse response of a room (conceived as a filter) is recorded and your signal is convolved with that impulse response in realtime by the reverb plugin.

Convolution

[Discrete convolution](#) is a mathematical operation that takes two discrete functions and generates a new function that multiplies every output of one function with every output of the other and sums them all. Convolution is expressed abstractly as:

$$(f * g)[n] \stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} f[m] g[n - m]$$

So, the function $(f * g)[n]$ ("f convolved with g of n") gives you a sum that represents multiplying

every output of the function $f[n]$ with the corresponding output of the function $g[n]$ and combining them all. The dependent variable 'n' determines how the outputs of the two functions are lined up, or which outputs of $g[n]$ are multiplied with which 'corresponding' outputs of $f[n]$. Typically we take $f[n]$ to be $x[n]$, the input to the system and $g[n]$ to be $h[n]$, the impulse response of a filter, so:

$$y[n] = \text{SUM}(x[m] \cdot h[n-m]) \quad \text{— (convolution of input } x[n] \text{ with filter } h[n])$$

A way to think about convolution is that you are generating a function that slides the output of two functions past each other, multiplying each element by each corresponding element and summing as you go along. The [wikipedia entry](#) has a good animated diagram for this.

Transfer Function

"Poles" (and "zeros") come from what's known as the [transfer function](#): $H(z) = Y(z)/X(z)$, which comes from the z-transform of the impulse response of the filter: $h[n]$. The transfer function expresses that the z-transform of the impulse response is equal to the z transform of the output ($y[n]$) of the system $x[n] \rightarrow [\text{Filter}] \rightarrow y[n]$ (the input convolved with the impulse response of the filter) *divided by* the z transform of the input ($x[n]$). This follows from the fact that multiplication in the "z-domain" is equivalent to convolution in the time domain. Just as the output of the system, $y[n]$ is equal to $x[n] * h[n]$ (remember, $*$ denotes convolution, not multiplication), in the z domain, the output of the system, $Y(z)$ is equal to $H(z)X(z)$ (the z-transform of the impulse response multiplied by the z-transform of the input).

Z-Transform

The z-transform is one of many different 'transforms' such as the famous [Fourier transform](#) and the [Laplace transform](#). The essence of a transform is that you are multiplying each output of a function onto a new "[basis](#)". The [canonical](#) example of a basis is the axes of a 3-dimensional plot: $[x, y, z]$. What's essential about a basis is that if we treat the axes x , y , and z as vectors (lines), you cannot 'get' one of the axes by adding one of the other axes with anything at all (a number or another line [vector] in the space). A basis is precisely a [vector](#) (line) that cannot be created by linear combination of other vectors. So, a shape in 3D space can be expressed by considering its position relation to the axes $[x, y, z]$. Likewise, a [signal](#) can be characterized by considering its value at a give time-step n . Just as $[x, y, z]$ is a basis onto which we can "project" 3D shapes, the set of discrete time steps n from $-\infty$ to $+\infty$ is a basis onto which we can project a function.

What 'transforms' do is they take a function and project it onto a new basis. Typically functions are projected onto "time" (n in discrete systems, t in continuous systems). The Fourier transform, on the other hand, projects a function onto an infinite array of sine waves of ever increasing frequency. This is accomplished simply by multiplication. If we want to project the function $f[n]$ into "Fourier space", we just do $\text{SUM}(f[n] \sin[n])$. *(However, please note that this is a naive explanation. The Fourier transform is _actually_ a projection onto complex functions of the form e^{in} . e^{in} is equivalent to $\cos[n] + i \sin[n]$. '*' denotes multiplication here.)*

The [z-transform](#) projects a function onto a gigantic polynomial. Remember, a polynomial is a function that looks like this:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0$$

Remember solving equations like this?

$$3x^2 + 4x - 5 = 0$$

Just as in the Fourier transform we project $f[n]$ onto an array of sine waves of ever increasing frequency, in the z-transform we project $f[n]$ onto an array of polynomial terms of ever increasing "degree". It looks like this: $\text{SUM}(f[n] \cdot z^{-n})$ or:

$$X(z) = \mathcal{Z}\{x[n]\} = \sum_{n=-\infty}^{\infty} x[n] z^{-n}$$

z^{-n} is a [complex number](#) $z = (y, x)$ raised to the negative n th power. The effect of doing the z-transform on a function is that you are projecting the output into the 2-dimensional "z-plane". By analyzing the response of the z-transform of the frequency response of a filter in the z-plane, we can learn about it.

Poles and Zeros

Poles and zeros come into the picture when we take a look at what the transfer function actually looks like in practice. From wikipedia, we learn that the transfer function can be massaged into a factored form where $H(z) = [\text{factored-polynomial-1}]/[\text{factored-polynomial-2}]$ like so:

$$H(z) = \frac{(1 - q_1 z^{-1})(1 - q_2 z^{-1}) \dots (1 - q_M z^{-1})}{(1 - p_1 z^{-1})(1 - p_2 z^{-1}) \dots (1 - p_N z^{-1})}$$

With the z-transform of the impulse response of the filter in question so represented, we can start to analyze the properties of the filter. Indeed, the reason that one uses the z-transform in the first place is that it is a particularly good representation for analyzing the properties of filters. One of the things that the transfer function shows, once it is in factored form, is when the equation will go to infinity and when it will go to zero. A point, or a value of z for which the equation is equal to infinity is called a pole. A point, or a value z for which the equation is equal to zero is called a zero. The location of poles and zeros can tell you a lot about the frequency response of a filter.

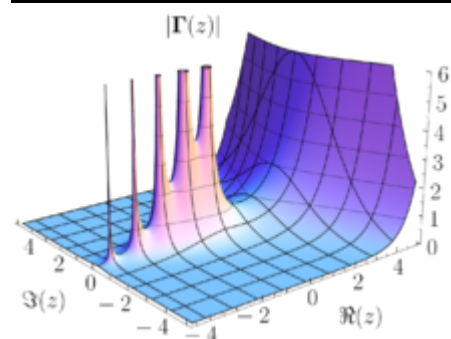
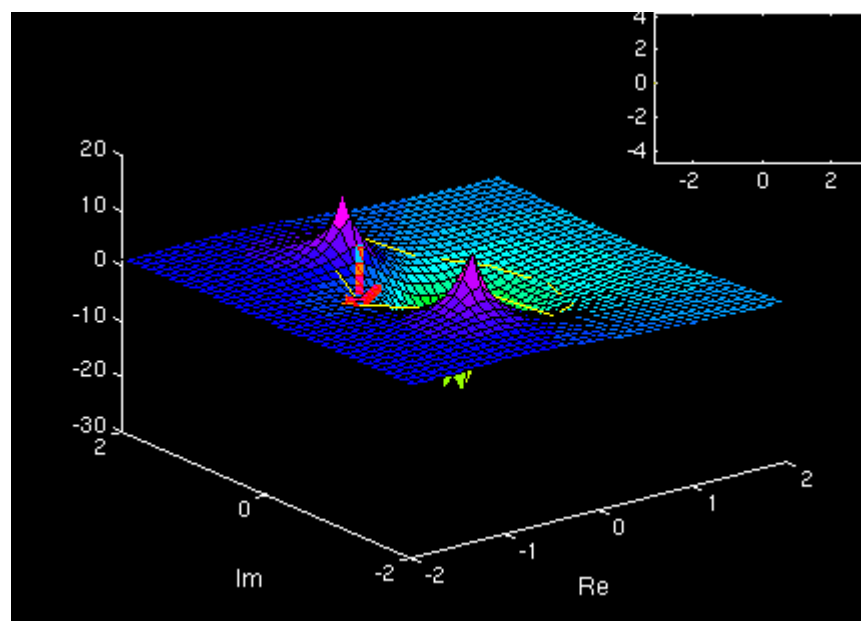
The Fourier Transform is a Subset of the Z-Transform

Before we can understand why, we must first note that the z-transform contains the Fourier transform. In fact, the Fourier transform is just the Z-transform evaluated on the [unit circle](#). The unit circle is a circle in the 2D complex number space that is centered around the origin, $z = (0,0)$ and has a radius of 1. Yes, this is the same unit circle from trigonometry. Position on the unit circle corresponds to the frequency response of the filter at that *frequency*. Frequency here is expressed in radians. So, if we evaluate $H(z)$ at $z = \pi/2 = 90$ degrees, we get an idea of what the filter is going to do to signals at that frequency. Frequency here is "normalized". 0 is the lowest frequency, π is the highest frequency, but π is not necessarily 20kHz. $-\pi$ corresponds to a negative frequency. In practice, negative frequencies are not reproduced by your speakers and

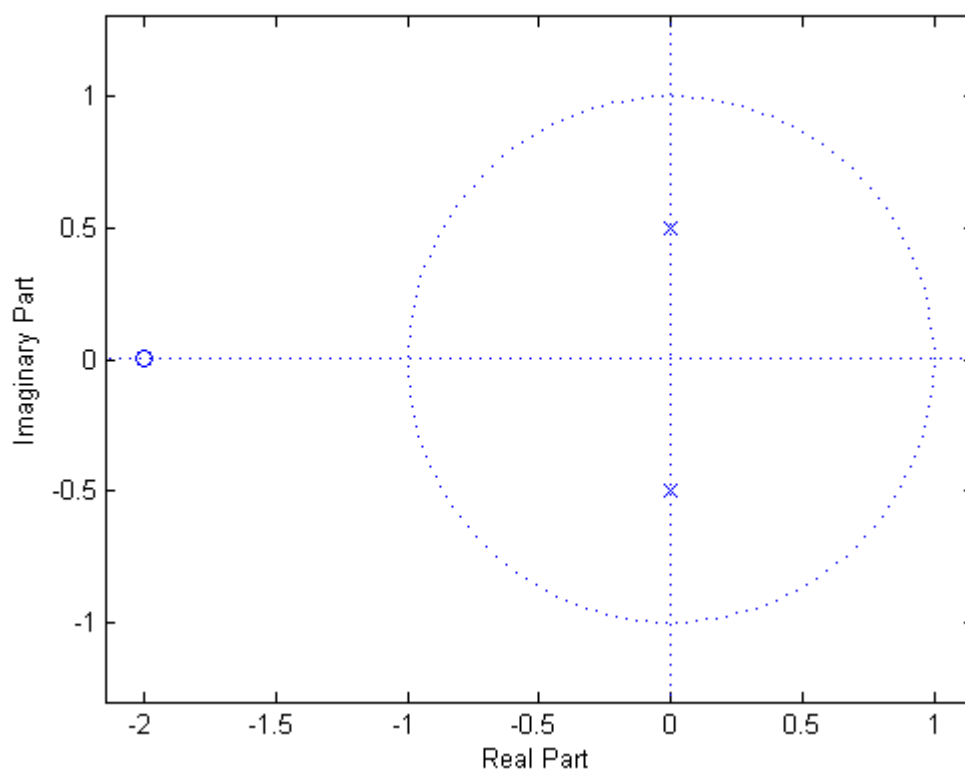
negative frequency. In practice, negative frequencies are not represented by your speakers and are ignored except for in the math. Let's, however, assume that $\pi = 20\text{kHz}$. If that's the case, then the response at $\pi/2$ corresponds to the response at 10kHz .

Pole Zero Analysis

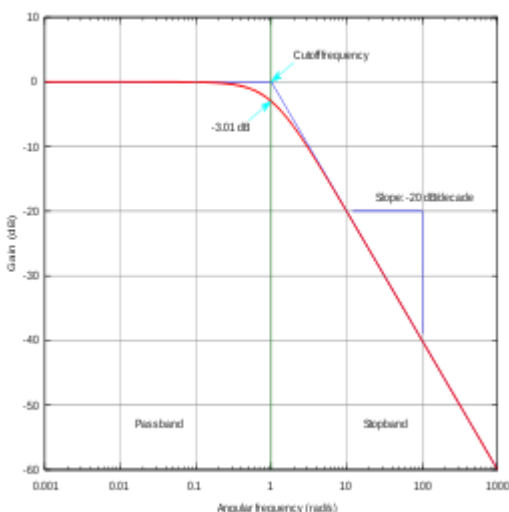
The reason that pole zero analysis is useful is that it lets us see the contours of the frequency response on the z-plane around the unit circle. A pole creates a peak that will amplify certain frequencies and a zero creates a valley that will attenuate certain frequencies. The z-transform is more useful than the Fourier transform for this kind of analysis because sometimes the poles and zeros are outside of the unit circle, in which case their effect on the frequency response is indirect. The best way to understand this is through a diagram:



And more conventionally we see [2D pole-zero plots](#) like this where x's are poles and o's are zeros.



Again, tracing the unit circle lets you see the response at different frequencies and the z-transform lets you see the mathematical properties of the function that influence the frequency response. Tracing the unit circle and projecting it into 2D Euclidian space gives you the familiar frequency response diagram:



Please correct me if I made any mistakes in my explanation. I'm not a signal processing student/professional – I've just taken one course in DSP.

edited Mar 3 '19 at 19:01



Mars

103 4

answered Dec 31 '10 at 20:10

msutherl


3 This is long and awesome! I'm going to bookmark this to read later... I suspect there's a lot to learn here :) –

Warrior Bob Dec 31 '10 at 20:31

That's way over my head, but +1 for the pics and formulae! – BenV Dec 31 '10 at 20:38

Morgan, tell me you didn't write this up at 8pm on New Years Eve. – Mtl Jan 29 '14 at 4:33

1 Joined the site just so I could +1 this answer! – norlesh Feb 3 '15 at 13:44

I can't believe *that* is the accepted answer when this exists. – underscore_d May 16 '16 at 16:39 

▲ Please read the Paragraph "Up the pole" on this site

1 <http://www.soundonsound.com/sos/sep99/articles/synthsecrets.htm>

answered Dec 24 '10 at 13:25

NES



8 This link would be more helpful if you summarized the contents here; especially since that link may not necessarily outlive this question – davetron5000 Dec 27 '10 at 13:55

I am not certain whether that is legal copyright-wise. – Pelle ten Cate Jan 3 '11 at 10:03

2 Summarizing in your own words is always perfectly legal. Copying small amounts is fair use. – endolith Jan 19 '13 at 6:07

1 Too late... The article has been removed, already. Luckily, however, I was able to fallback to Web Archive: web.archive.org/web/20160506143719/http://www.soundonsound.com/... – Rui Pimentel Nov 30 '16 at 0:33