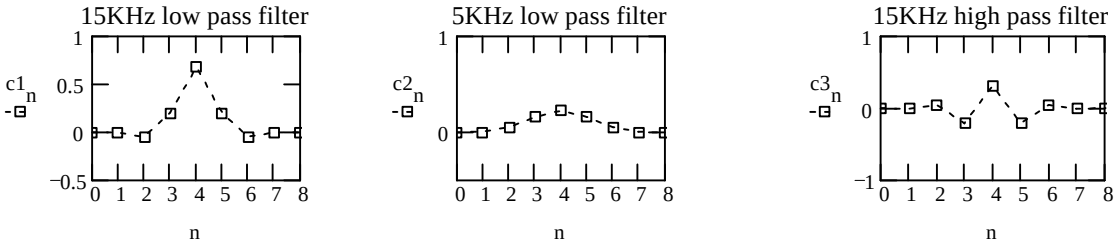
Understanding FIR (Finite Impulse Response) Filters - An Intuitive Approach
by Dan Lavry, Lavry Engineering

People less familiar with digital signal processing, often view the theory as "incomprehensible" and the hardware as "little black boxes". This article is aimed at replacing total mystery with some "feel" and "common sense understanding". While not easy reading, a little patience and some concentration will shed much light regarding the "mechanics" of processing a
signal by simple means such as multiplication and addition.

Introduction

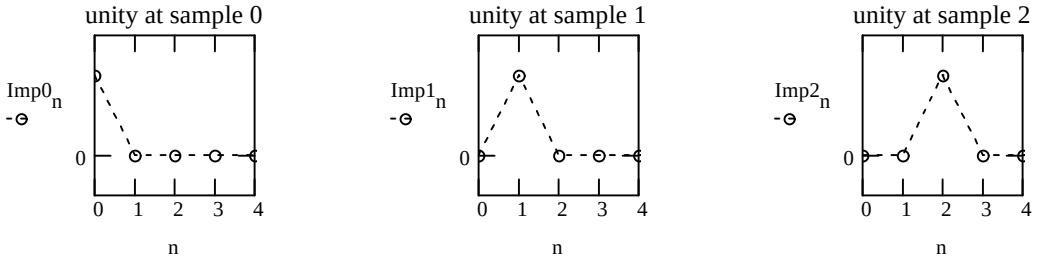
Digital audio signals are expressed in terms of sample values, equally spaced in time. Filters based on a rather simple "computational structure" known as an FIR, yields impressive results. FIR filter frequency response is determined by a set of coefficients. Bellow are examples of three different sets of coefficients:



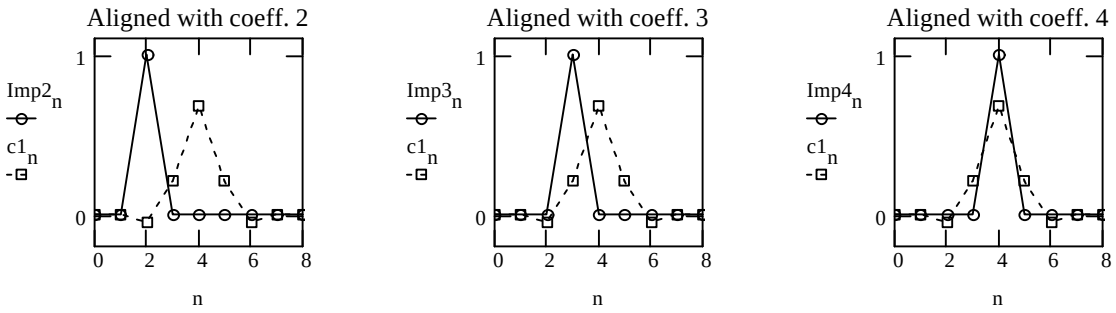
The three examples, each consisting of 9 coefficients (0 through 8) show three different coefficient curves. The filter uses only the numerical values graphically shown as "boxes". The zero coefficient for the 15KHz low pass has a value 0. coefficient 3 is .208, coefficient 4 is .68 and so on.

Impulse response

The purpose of the coefficients is to alter the signal content by means of simple arithmetic. The simplest case to demonstrate is the response of a filter to impulse. An impulse waveform has zero amplitude at all but one the sample points. Each sample takes the nonzero value sequentially (one sample at a time). The plots bellow show the propagation of an impulse from "sample 0" to "sample 1" to "sample 2".



Let us "pass" the impulse signal through a filter. We use the 9 coefficient (0 through 9) 15KHz low pass shown above. Input signal sample values are shown by circles, and the coefficient value as a squares:



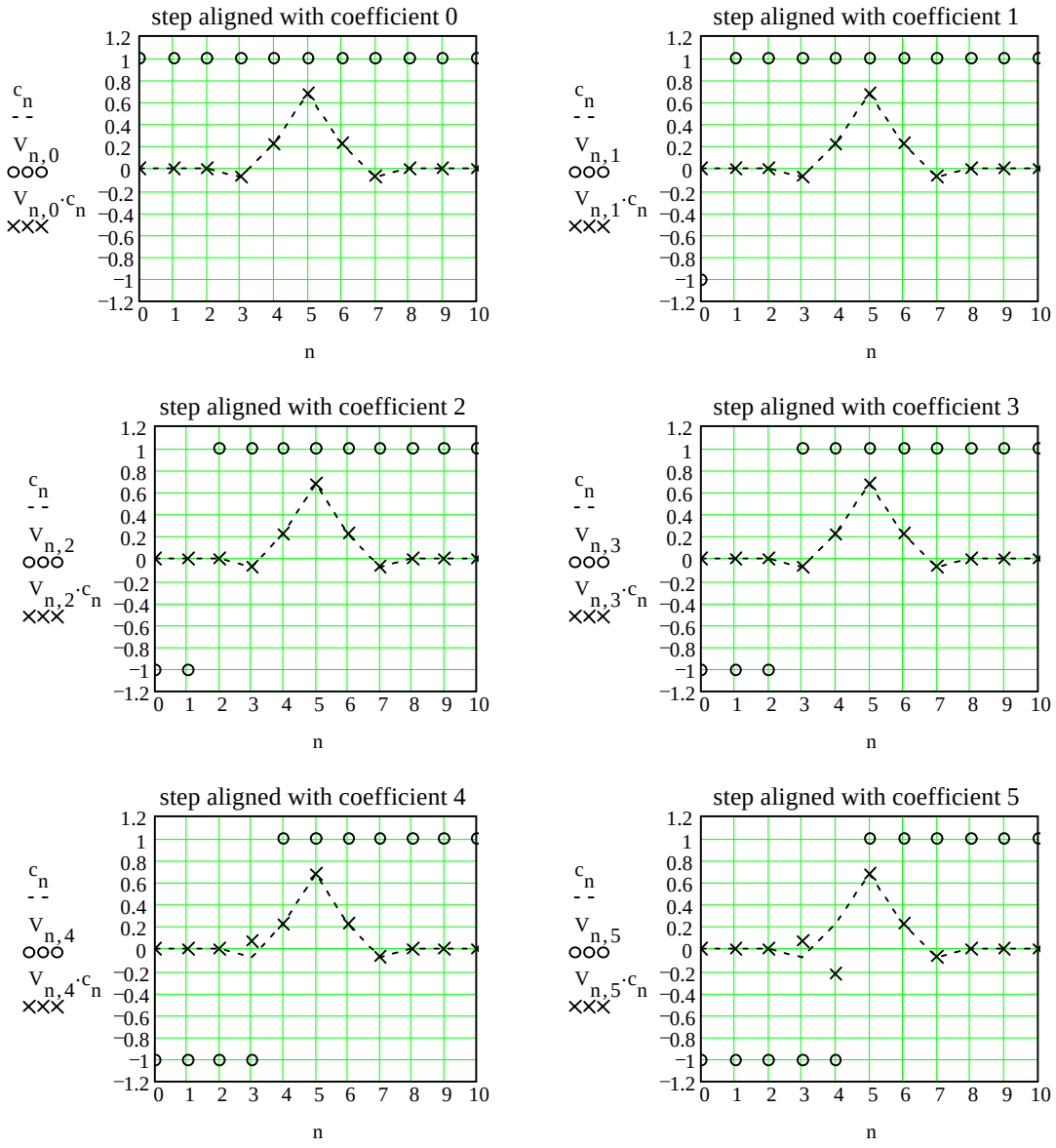
The impulse "travels" from left to right, against a "fixed frame" of coefficients. Each plot shows a different instant in time, enabling computation of one filter output sample. The calculation is a two step process:
1. Multiply the values of each pair (samples represented by circles and coefficient represented by a squares). We get nine new values known as products.
2. Add the nine products. The sum of the products is the value of the corresponding **output sample**.

Clearly, an impulse input with amplitude 1 will generate an output waveform identical to the shape of the coefficient plot. The "walking" impulse has 0 value at all but one points at a time. The coefficients get aligned with the unity impulse, one at a time, while the rest are aligned with zero's. The sum of the products will reproduce the shape of the coefficients curve. The FIR output yields a finite number of non zero values, thus the name Finite Impulse Response.

Real life signals are more complex, and the multiplications yield many non zero values, but the process of alignment, sample - coefficient pair multiplication and final summation remains the same. We next examine the "next to simplest case".

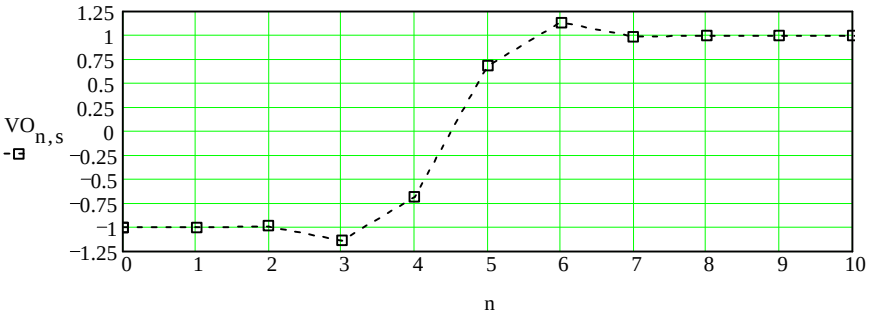
Step response

A step response input propagates an "instantaneously rising" signal through the filter. The signal steps from -1 to +1. The six consecutive "frames" below show the signal rise alignment with coefficients 0 through 5. The x symbols denotes the products of input values (circles) and coefficient values (expressed by the dotted line).

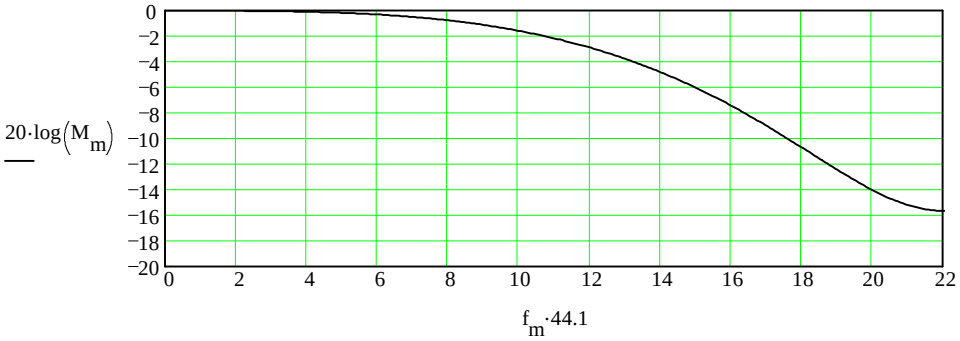


The step response requires very simple arithmetic. All the coefficients to the left of the step are multiplied by an input signal values of -1 (inverting the coefficient curve to the left of the step). The coefficients to the right of the step are multiplied by 1, leaving the curve unchanged. Examine samples 3,4,5 and 6 of the right bottom plot.

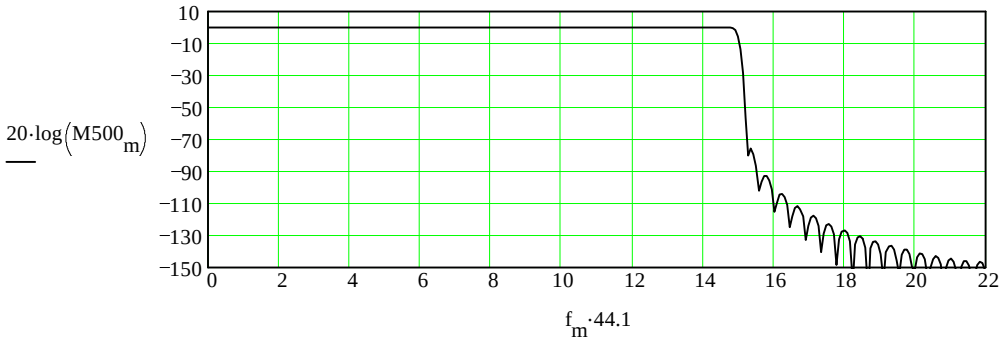
Calculating output sample value for each "frame" requires summing (accumulating) all 9 products. Adding the positive and subtracting the negatives values (shown by the x's) yield the output sample value. The "next" output sample is determined by shifting the signal - coefficient alignment by one and recomputing the sum of the products.
The filter used in the above example is a 15KHz low pass (at 44.1KHz sampling rate). The plot below shows 10 output samples corresponding to 10 frames. The slower rise and the overshoot are due to attenuation of the harmonics above 15KHz.
Wider coefficients "bell shaped" curves would modify the input step over more output samples, slowing the rise, thus lowering filter cutoff frequency.



The frequency response of the filter is shown bellow. Incoming frequencies below, say 4KHz are left unattenuated. The attenuation at 20KHz is about 14dB.

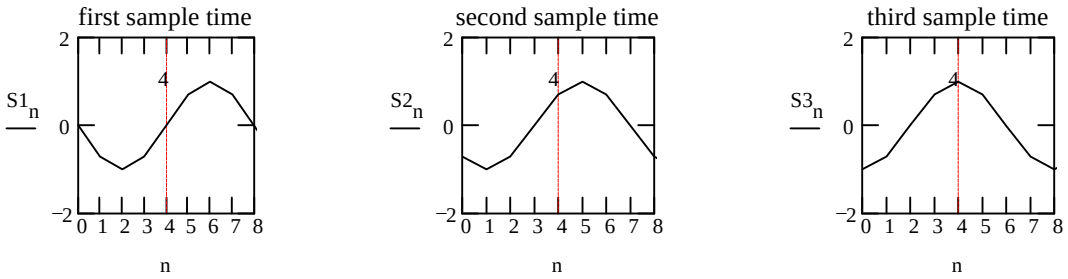


Let's examine a response of a 500 coefficient 15KHz low pass filter. The response is flat to 15Khz. The attenuation is 100dB at 16Khz and 140dB at 22Khz. A 500 coefficient filter requires 500 multiplication's and 500 additions for each output sample. Specialized integrated circuits (digital signal processors) can often exceed such enormous computational requirement. The the plot shows the value of such a compute engine. Old analog design technology does not yield such results. Other FIR advantages will be discussed later.

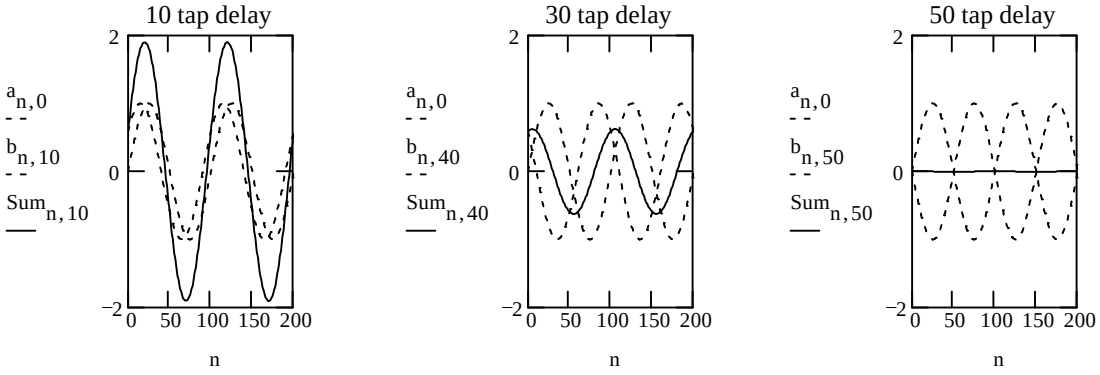


Sinusoidal waves and periodic waveforms

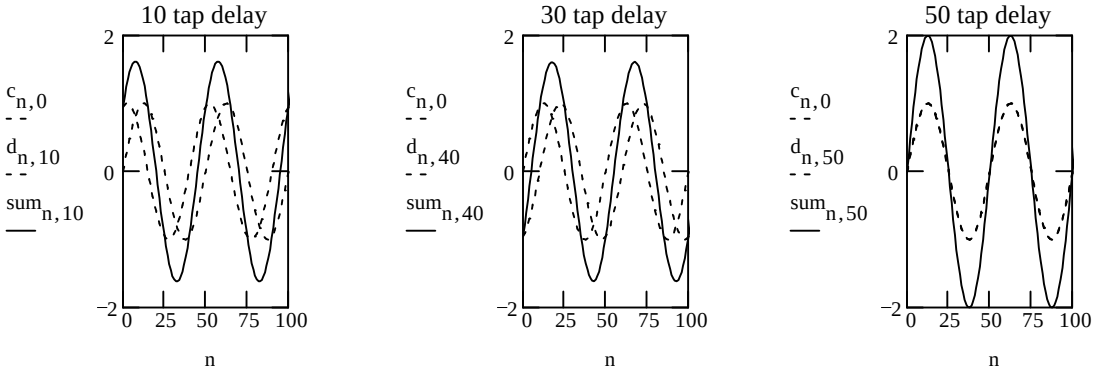
The multiply and accumulate processes can be done for any waveform. To gain further insight, lets view the case of a sine wave filtering from a somewhat different angle. Impulse and step response inputs "walk" a single transition through the filter, and each coefficient is aligned with the transition only once. When "walking" a sine wave through a filter, each of the coefficients finds itself lined up with a sine wave (the same sine wave samples move from one coefficient tap to another. The 3 frames below show the sine wave motion. Tap number 4 receives 0 at first sample time, .707 value at the second and 1 for the third. Continuing the process will present tap number 4 with a periodic set of values, following the input sine wave. The same is true for all the taps. The difference between taps is only the "time of arrival" of the signal (or time delay).



Allow me to express an interesting fact (proof can be found in basic trigonometry): Adding sine waves, all of the same frequency, produces a sine wave. In other words, while the amplitude and delay of the sum depend on the amplitude and delay of the individual components, the wave form of the sum remains a sine wave. When an FIR filter adds a number of delayed sine waves, each component has its own amplitude (each delay tap has its own multiplying coefficient), but the outcome remains a sine wave. Thus an FIR causes no harmonic distortions.
The following plots demonstrate how the sum of two equal amplitude 4.4KHz waves (sampled at 44.1KHz) yield different outcomes. With 10 sample delay, the signals are almost in phase, thus the amplitude is almost doubled. With 50 tap delay, the signals cancel each other and the output is attenuated to zero.



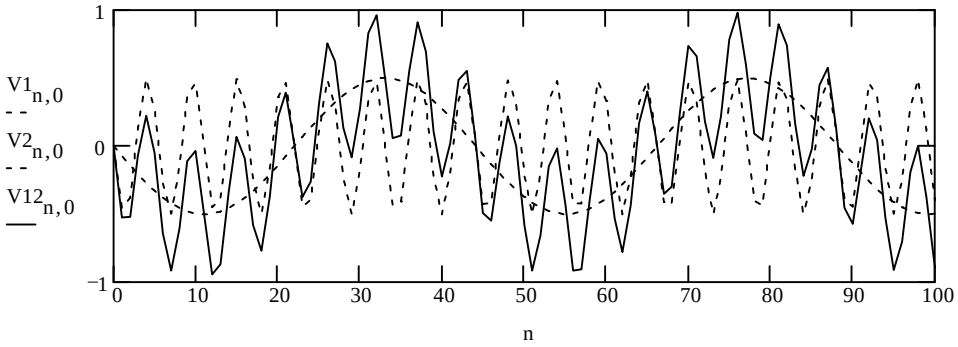
Let us repeat the plots for an 8.8KHz input signal, with the same time delays:



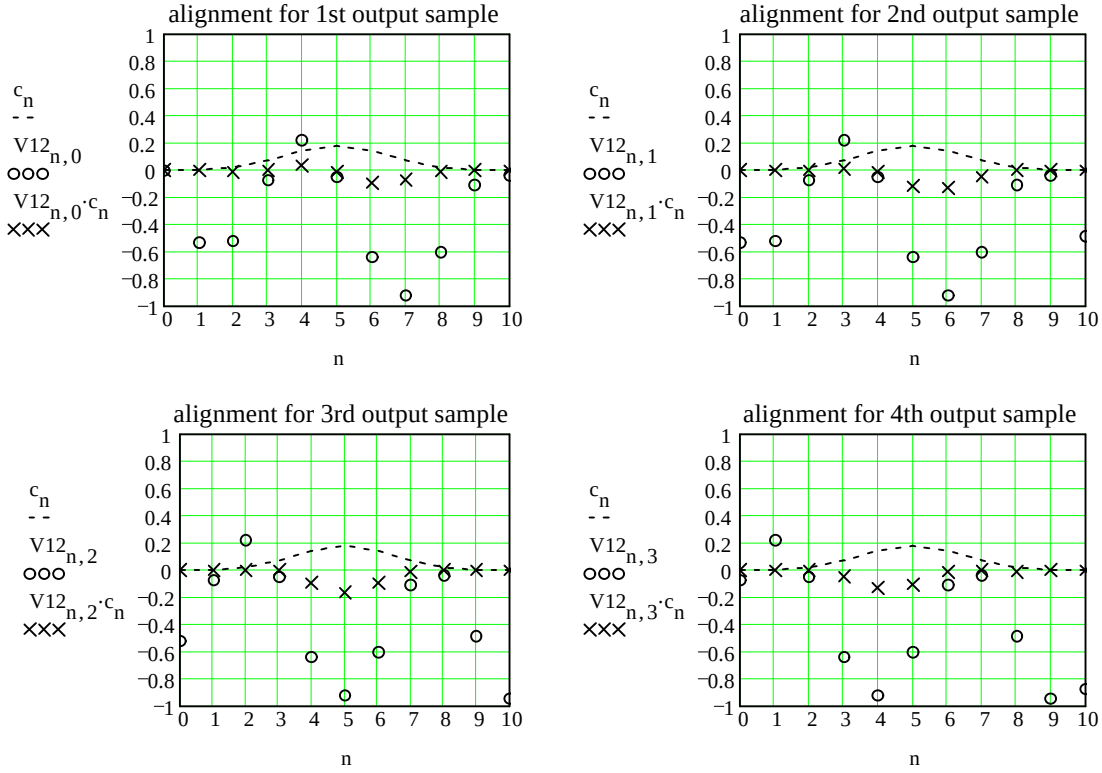
Comparing the "filters outputs" at 8.8KHz to the 4.4Khz plots show that output amplitudes depend on input frequencies. The 50 tap delay shows an extreme case (double the amplitude at 8.8KHz and complete attenuation at 4.4KHz).
The above filters are very simple, yet they serve to demonstrate the basis for sine wave filtering. Real world FIR's are made of numerous taps and with coefficients that are rarely equal to 1, enabling great control over filter response.

What about periodic waves? All periodic waveforms are, in essence, a sums of basic sine waves components (Fourier series). Feeding an FIR with any wave shape is the same as sending the individual components through the filter. The FIR filters a complex periodic waveform as if it were operating on each sine wave component separately (independently). Multiplying a sample by a coefficient yield the same result as "breaking" a sample into parts, multiplying each part by a coefficient and summing the products. The concept is referred to as a "linear system". Filter behavior across the frequency spectrum is completely defined by it behavior at each input frequency (one at a time).

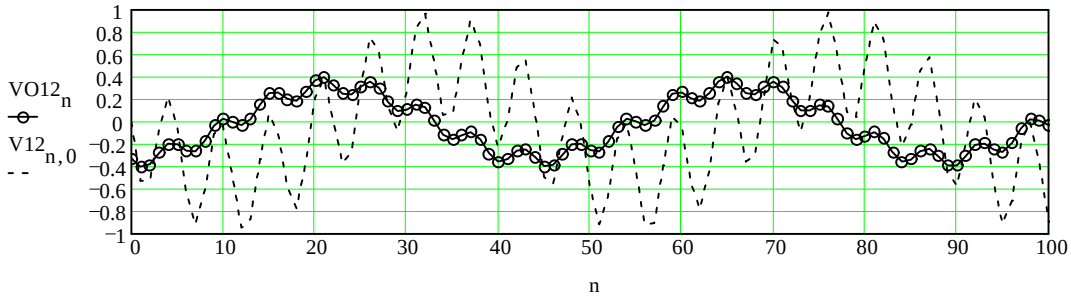
Let us demonstrate the workings of an FIR with a two tone example. The plot shows 1KHz and 8KHz tones (at 44.1KHz sampling), with dotted lines. The filter input V12 (sum of V1 and V2) is shown by a solid line.



Let's filter the first four points of the dual tone signal with a 10 tap 3KHz low pass coefficients. The four frames show a "fixed" coefficient curve (dotted lines), the signal (circles) moving through the filter (by one sample per frame from right to left). The individual product pairs are marked by x symbols.

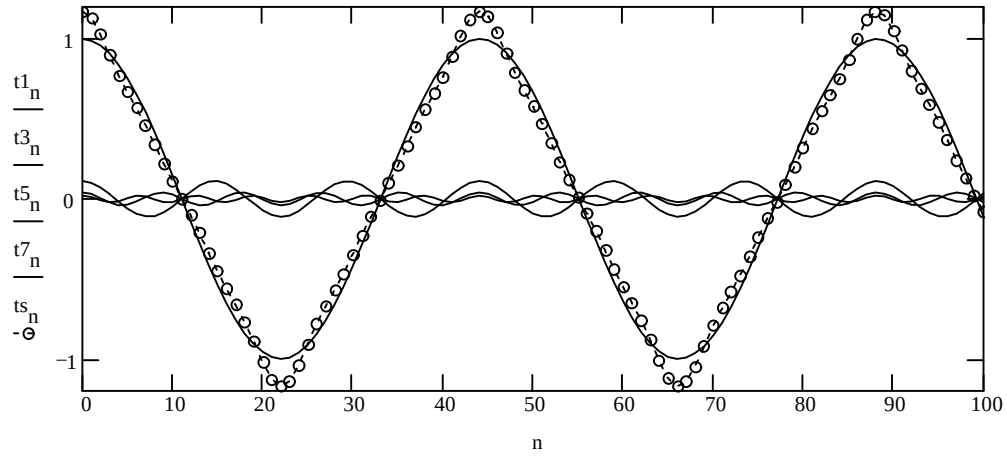


Accumulating the x vales for each of 100 frames (four of which are shown above) yield the output samples (circles in the plot bellow) .The dotted line is the input. Note the high frequency attenuation:

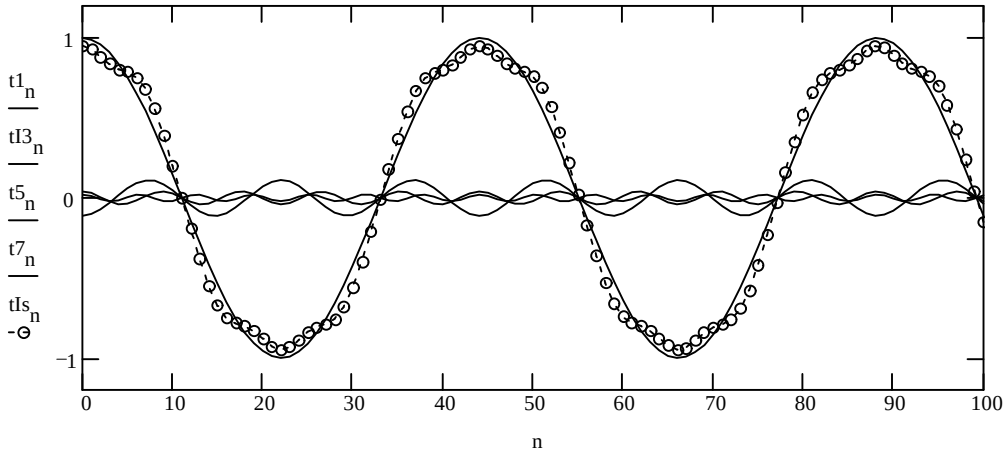


Phase linearity

An important FIR advantage is phase linearity. Most filters (analog and digital) introduce different amount of delay for different frequency components of the signals. The FIR keeps the delay for all frequency components the same.
Let us examine the outcome of varying the delay of frequency components. We start by approximating a 1KHz triangle shape wave by adding 1, 3, 5 and 7KHz components of appropriate amplitude and phase relationships. Notice that the four sine wave components peak together and add up to maximize the triangle wave peak point (the circles shows the sum):



If we invert the third harmonic (delay it by half a cycle time) it's contribution at "peak time" is inappropriate as shown bellow. The harmonics no longer "peak together" and the outcome is far from triangle.



While somewhat crude, the example serves to demonstrate that wave shape retention requires constant time delay at all frequencies. How important is the shape of the wave? The answer depends on the application. A frequency dependent delay is analogous to having different distance from sound source to the ear for different frequency components. The impact is not very noticeable with a monophonic sound source, but rather dramatic for stereo or higher number of channels.
Phase nonlinearities (frequency dependent delay) is considered "enemy number one of proper sound field imaging" by many recording engineers. FIR's do not introduce such problems.

The coefficients curve:

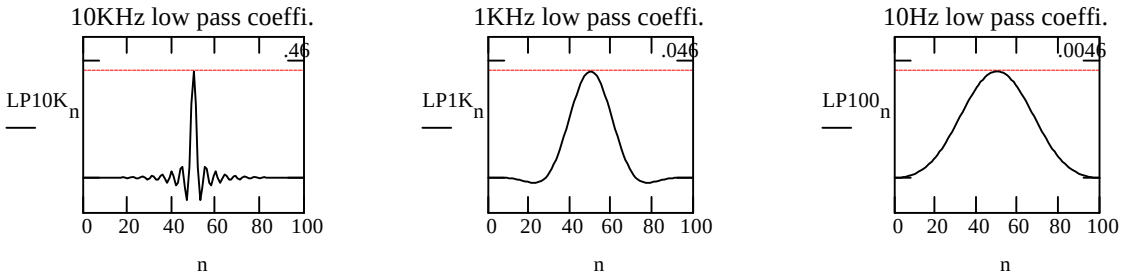
Most filter designers use a "ready made software". Others (including the author of this article) often go back to mathematics and digital signal processing literature. Understanding the relationships between coefficients and filter performance requires some math, yet some aspects remain intuitive:

DC characteristics: when feeding an FIR filter a DC signal, all filter taps receive a constant value. Let us assume a DC value of 1. The sample - coefficient product at each tap equal the coefficient value (times 1). Each output sample becomes the sum of the coefficients. Setting the coefficients sum to 1 makes a filter with unity gain at DC. Generally speaking, one can scale the coefficients to adjust gain. Let us restrict all further comments to unity gain filters (gain scaling is rather straight forward). All low pass and notch FIR filters must pass DC thus their coefficients sum is 1. Blocking DC (high pass, band pass or other DC blocking filters) requires a coefficient sum of 0.

Relative value of the coefficients: large value coefficients carry more weight in the construction of output samples. Small value coefficient serve to "fine tune" the filter response. 120dB attenuation require fine details of 1 part per million.

The number of required coefficient: (or the width of the coefficients curve): An "all pass" filter consists of one tap (each sample value is multiplied by 1 and is sent to the output). One input sample is insufficient for any other filter response. Filtering requires examining a number of samples around the point of interest. Increasing the number of samples (filter taps) yield more accurate information. A very low frequency filter receives a slowly changing signal, thus requiring the designer "to view" signal behavior many samples away.

The three low pass filters bellow consists of 101 coefficients (0 to 100). The 10KHz filter low pass relays heavily on the "center" coefficients. "tap number 50" contributes almost 1/2 of the output value. The values away from center approach zero value fast. The 100Hz filter curve is significantly wider in shape. The largest coefficient contributes less then .5% of the final outcome, and taps further from center play a significant role.



Closing remarks:

Integrated circuits for digital signal processing (DSP) are a specialized breed of computational engines designed, for the most part, to simultaneously move sampled data from tap to tap, while computing very large numbers of multiplications and additions. Processing a single 44.1KHz channel with a 500 tap filter requires 88.2 million computations per second. Despite many clever schemes for increased computational efficiency, a compromise between desired response and the number of taps is not uncommon. The tradeoff is between attenuation, flat response, ripple in the passband (and attenuation region), transition band(s) and more. Other compromises have to do with computational accuracy. The number of digits (bits) available for both coefficients and input signals may plays a major role in filter quality. Each sample and each coefficient is dealt with to a finite hardware dependent accuracy. The accumulated errors may take a toll on the final output result. At times, adding taps may cause more harm then good. The filter designer should take all such factors into consideration. Yet, the FIR offers a viable solution over much of the frequency range. Shaping the frequency response at very low frequencies or near Nyquist (half the sampling frequency) often requires prohibitive amount of compute power, and use of another filter type: the IIR. Analog filters and IIR's yields similar frequency and phase characteristics, yet analog relays on component values (such as resistors and capacitor). FIR and IIR filters use arithmetic.