Sampling Theory For Digital Audio

By Dan Lavry, Lavry Engineering, Inc.

Credit: Dr. Nyquist discovered the sampling theorem, one of technology's fundamental building blocks. Dr. Nyquist received a PhD in Physics from Yale University. He discovered his sampling theory while working for Bell Labs, and was highly respected by Claude Shannon, the father of information theory.

Nyquist Sampling Theory: A sampled waveforms contains ALL the information without any distortions, when the sampling rate exceeds twice the highest frequency contained by the sampled waveform.

Introduction

While this article offers a general explanation of sampling, the author's motivation is to help dispel the wide spread misconceptions regarding sampling of audio at a rate of 192KHz. This misconception, propagated by industry salesmen, is built on false premises, contrary to the fundamental theories that made digital communication and processing possible.

The notion that more is better may appeal to one's common sense. Presented with analogies such as more pixels for better video, or faster clock to speed computers, one may be misled to believe that faster sampling will yield better resolution and detail. The analogies are wrong. The great value offered by Nyquist's theorem is the realization that we have ALL the information with 100% of the detail, and no distortions, without the burden of "extra fast" sampling.

Nyquist pointed out that the sampling rate needs only to exceed twice the signal bandwidth. What is the audio bandwidth? Research shows that musical instruments may produce energy above 20 KHz, but there is little sound energy at above 40KHz. Most microphones do not pick up sound at much over 20KHz. Human hearing rarely exceeds 20KHz, and certainly does not reach 40KHz. The above suggests that 88.2 or 96KHz would be overkill. In fact all the objections regarding audio sampling at 44.1KHz, (including the arguments relating to pre ringing of an FIR filter) are long gone by increasing sampling to about 60KHz.

Sampling at 192KHz produces larger files requiring more storage space and slowing down the transmission. Sampling at 192KHz produces a huge burden on the computational processing speed requirements. There is also a tradeoff between speed and accuracy. Conversion at 100MHz yield around 8 bits, conversion at 1MHz may yield near 16 bits and as we approach 50-60Hz we get near 24 bits. Speed related inaccuracies are due to real circuit considerations, such as charging capacitors, amplifier settling and more. Slowing down improves accuracy.

So if going as fast as say 88.2 or 96KHz is already faster than the optimal rate, how can we explain the need for 192KHz sampling? Some tried to present it as a benefit due to narrower impulse response: implying either "better ability to locate a sonic impulse in space" or "a more analog like behavior". Such claims show a complete lack of understanding of signal theory fundamentals. We talk about bandwidth when addressing frequency content. We talk about impulse response when dealing with the time domain. Yet they are one of the same. An argument in favor of microsecond impulse is an argument for a Mega Hertz audio system. There is no need for such a system. The most exceptional human ear is far from being able to respond to frequencies above 40K. That is the reason musical instruments, microphones and

Copyright Dan Lavry, Lavry Engineering, Inc, 2004

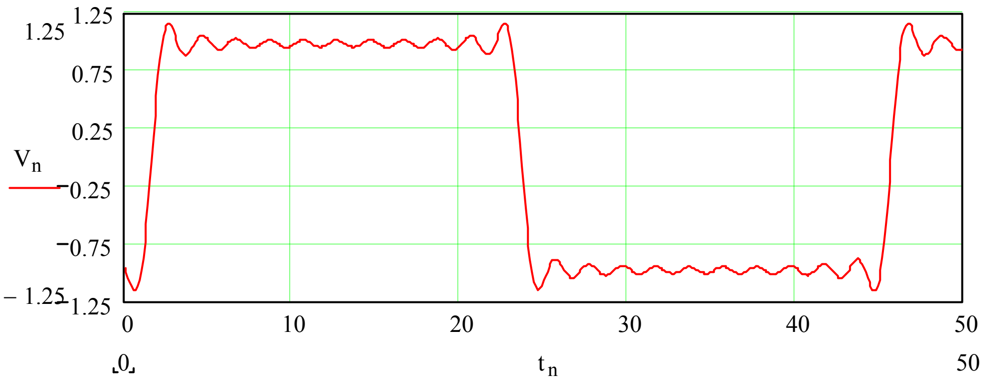
speakers are design to accommodate realistic audio bandwidth, not Mega Hertz bandwidth.

Audio sample rate is the rate of the audio data. Such data may be generated by an AD converter, received and played by a DA converter, or even altered by a Sample Rate converter.
Much confusion regarding sample rates stems from the fact that some localized processes happen at much faster rates than the data rate. For example, most front ends of modern AD (the modulator section) work at rates between 64 and 512 faster than a basic 44.1 or 48KHz system. This is 16 to 128 times faster than 192KHz. Such speedy operation yields only a few bits. Following such high speed low bits intermediary outcome is a process called decimation, slowing down the speed for more bits. There is a tradeoff between speed and accuracy. The localized converter circuit (few bits at MHz speeds) is followed by a decimation circuit, yielding the required bits at the final sample rate.

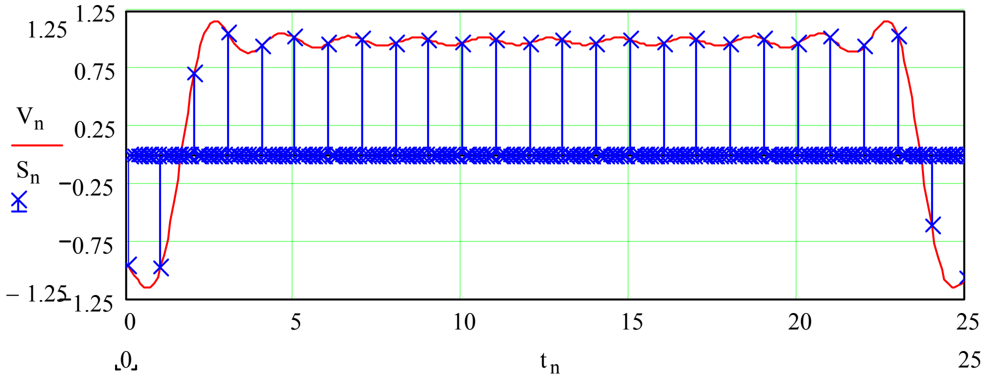
Both the overall system data rate and the increased processing rate at specific locations (an intermediary step towards the final rate) are often referred to as “sample rate”. The reader is encouraged to make a distinction between the audio sample rate (which is the rate of audio data) and other sample rates (such as the sample rate of an AD converter input stage or an over sampling DA’s output stage).

Sampling

Let us begin by examining a band limited square wave. We set the fundamental frequency to 1KHz and the channel bandwidth to 22.05KHz (as in red book audio CD). A quick calculation yields 22 harmonics (though for a square wave all even harmonics are zero amplitude). The plot below shows the addition of the 22 harmonics:

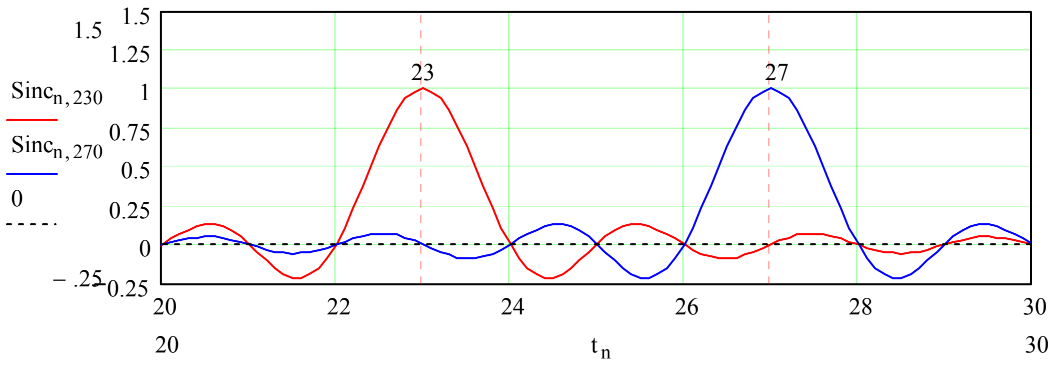


Let us magnify the part of the wave (red) between t=0 and t=25. The blue lines define sample times. The blue X's show the value assigned to each sample. For example, the value of the sample at t=1 is about -1, at t=2 we have .75 and so on. Our sampled wave can be represented by a sequence of values



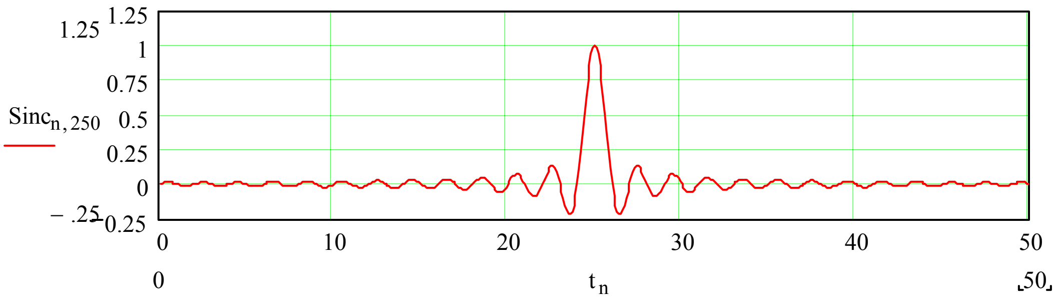
Initial intuitive reaction may cause one to think that we do not have enough X's to be able to re plot the original wave (red) with all of its details. That intuitive reaction is wrong. The key here is fact that the wave form is band limited. For a given bandwidth, the number of samples (X's) need only to exceed twice the bandwidth in order to be able to retrieve the complete waveform, including any value between the sample times. Let us see how it is done.

We are now going to introduce our sampling sinc functions. For the mathematically inclined sinc(x)=sin(x)/x. Let us plot two sinc functions, one centered at 23 (red) and the other at 27 (blue). Each sinc plot consists of a "main lobe" and ringing (decaying sin wave) on each side of the main lobe. Note that each sinc crosses zero (the dotted black line) at t=20, 21, 22, 23... All our sampling sinc functions will be positioned to have zero (zero crossing) at the same points. Note that the ringing between the peaks of the two sincs (between 23 and 27) goes in opposite directions. The ringing below 22 and above 28 is in the same direction.

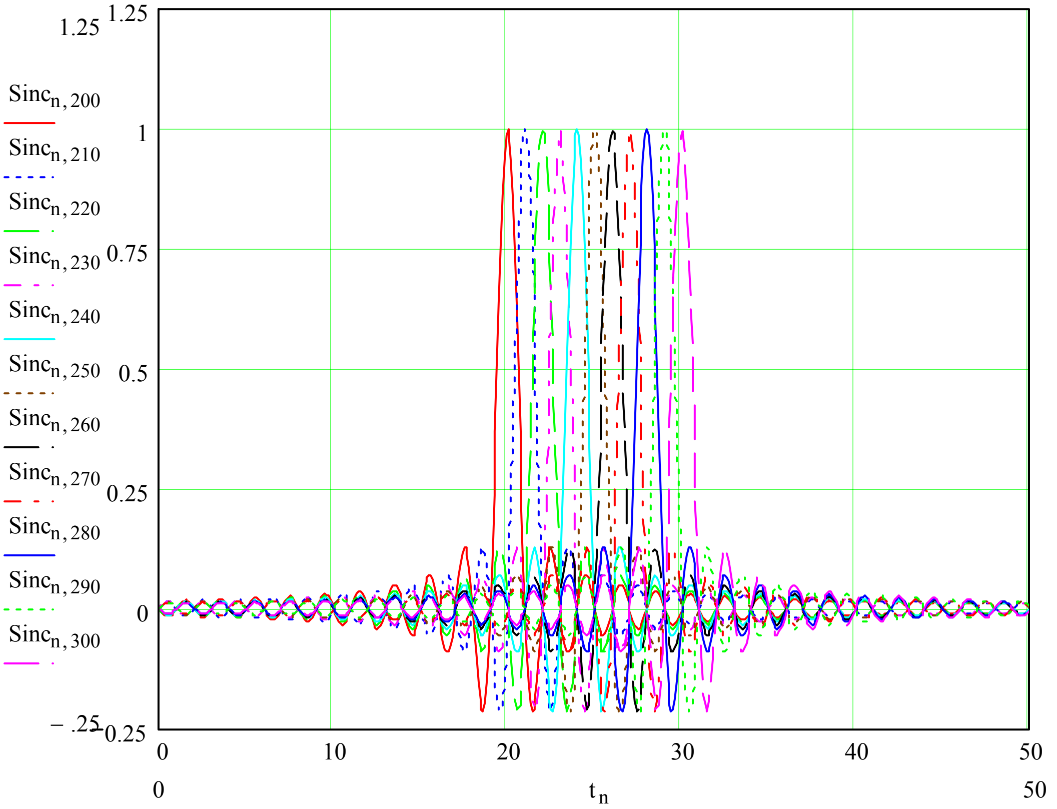


We can see that the sinc plot continues to "ring" (go up and down) beyond the plot limits selected (n=20 to n=30. When we expand the range we see that the "ringing" continues, but gets smaller in amplitude as we get away from the center. If we go far enough, it approaches zero and gets negligible for all practical purposes.

Copyright Dan Lavry, Lavry Engineering, Inc, 2004

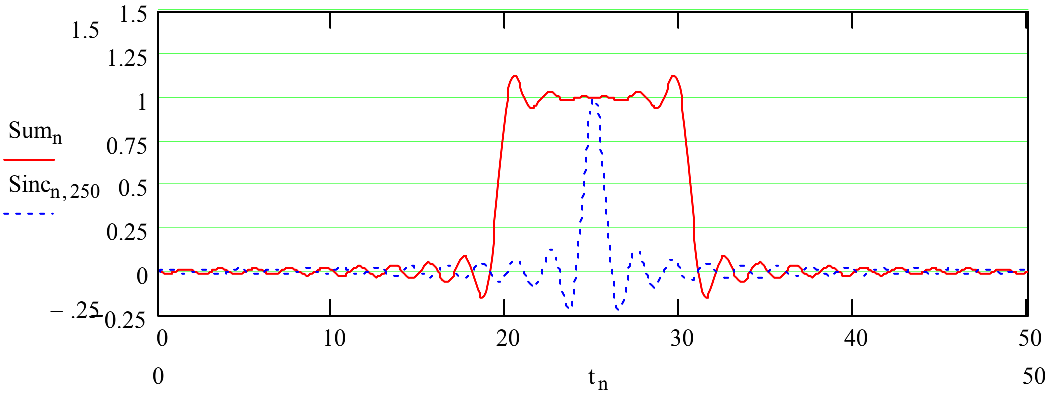


Let us now generate 11 equal amplitude sinc functions spaced apart. We center them at t=20,21,22...30.

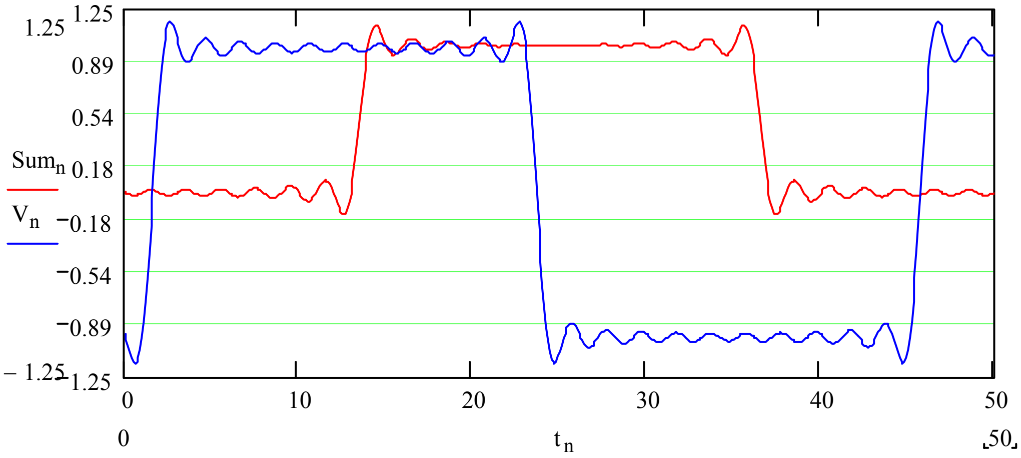


Adding the above 11 equal amplitude sinc functions yields the pulse shape below (red). Note that we are beginning to see some similarity to a band limited square wave. The blue dotted plot is a single sinc for visual reference.

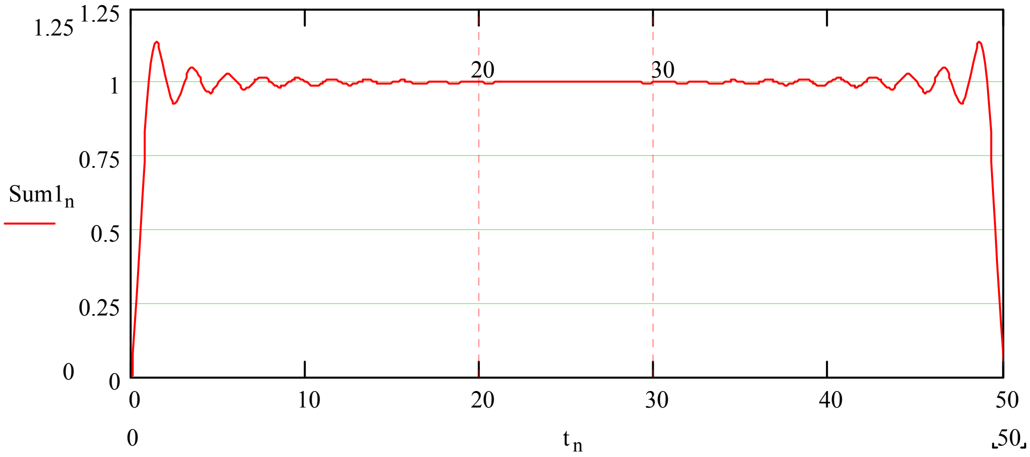
Copyright Dan Lavry, Lavry Engineering, Inc, 2004



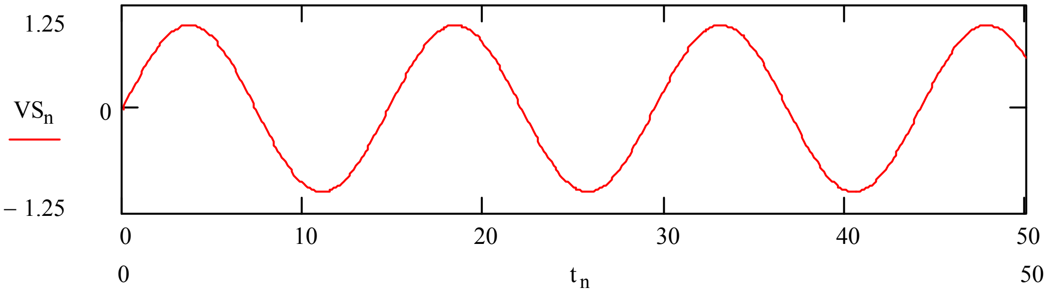
Let's add 22 adjacent sincs. 22 because this will yield a half cycle width of a 1KHz square wave. The red wave bellow shows the outcome - a positive half of a band limited square wave. The blue wave is the band limited 1KHz square wave we constructed by adding 22 sine waves. (see the first plot in this paper). Note the similarity of the positive half of the waves (ignoring the horizontal shift in time).



Can one generate DC using such “twisted” sinc functions? The plot below is a sum of 480 equal amplitude sinc's. As mentioned earlier the width and position of the sinc is set so that the peak of the "main lobe" and the zero crossings of the ringing occur at sample time. Note that the center of the sum (t=20 to t=30) is a good approximation of a DC signal (value 1). Each individual sinc is far from a straight line, but their sum can yield DC. In fact, we can find a set of sinc functions such that when adding them will yield any band limited wave shapes we desire.

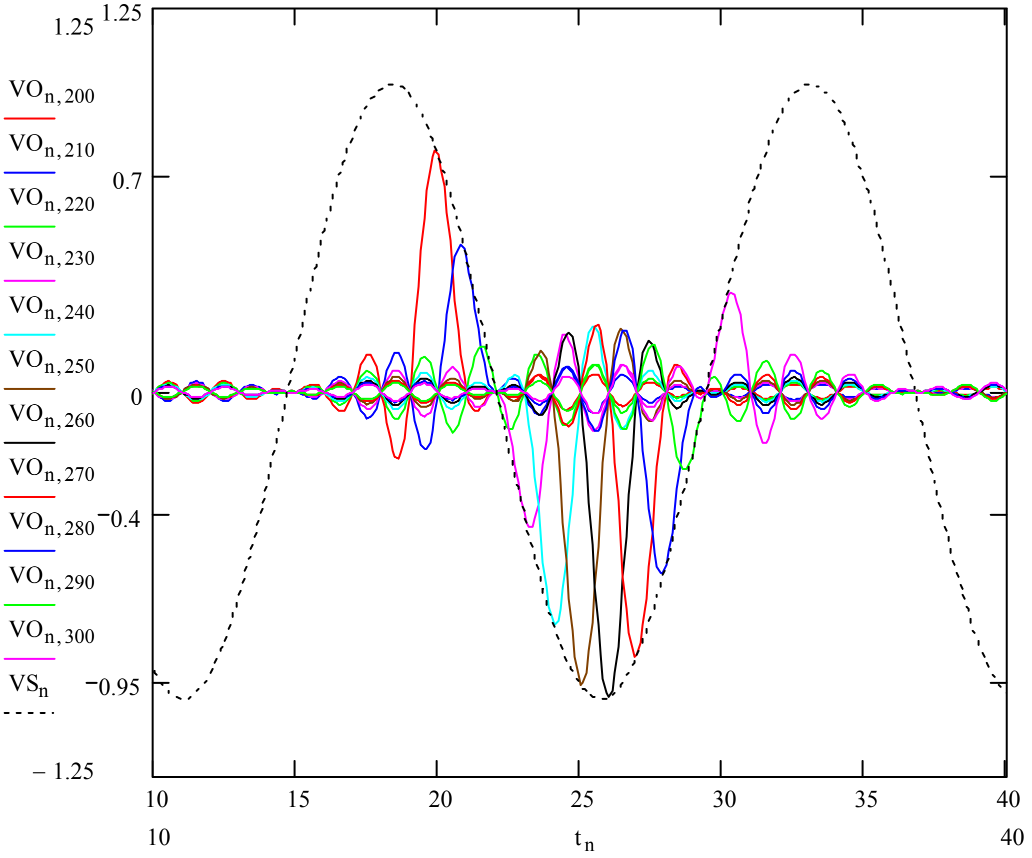


Let us find the set of sinc functions that correspond to a 3KHz sine wave. We begin by plotting the wave:

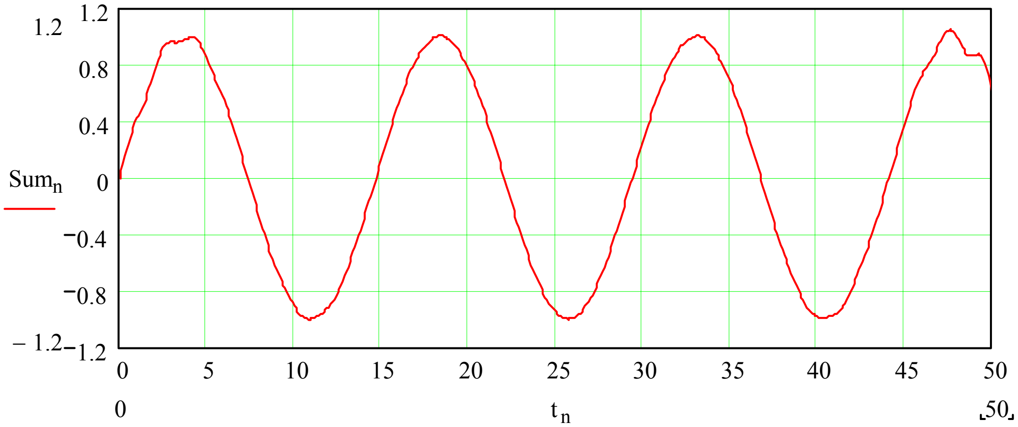


Next we multiply each sample point by a sinc function. The plot below shows the multiplication of the 3KHz tone (dotted black) by a sinc at 11 locations (20 to 30). The center of each sinc is aligned and adjusted to the amplitude the sine wave at each sample time. Note that the sinc function gets inverted for negative values of the sine wave.

Copyright Dan Lavry, Lavry Engineering, Inc, 2004



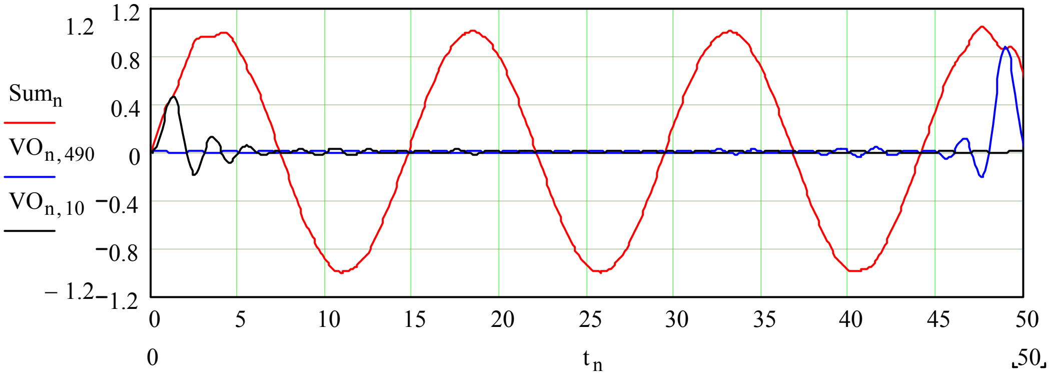
With sincs positioned and multiplied by the input wave at ALL sample points, we are now ready to see what the combined outcome is. Adding all sinc's yields the wave below.



Note that the first half cycle (t=0 to 7) and the last half cycle (t=45 to t=50) are distorted. The middle (t=7 to t=45) looks similar to the 3KHz sine wave input. The distortions at both ends are due to an abrupt start and stop of the wave. Any sudden beginning or ending amounts to high frequency content. A sine burst itself contains a sine wave and a gating wave (turning the sine on and off). The gate requires infinite bandwidth. A different way to state this is that the sinc functions at the beginning and end are truncated; thus their contribution is partial (see next plot).

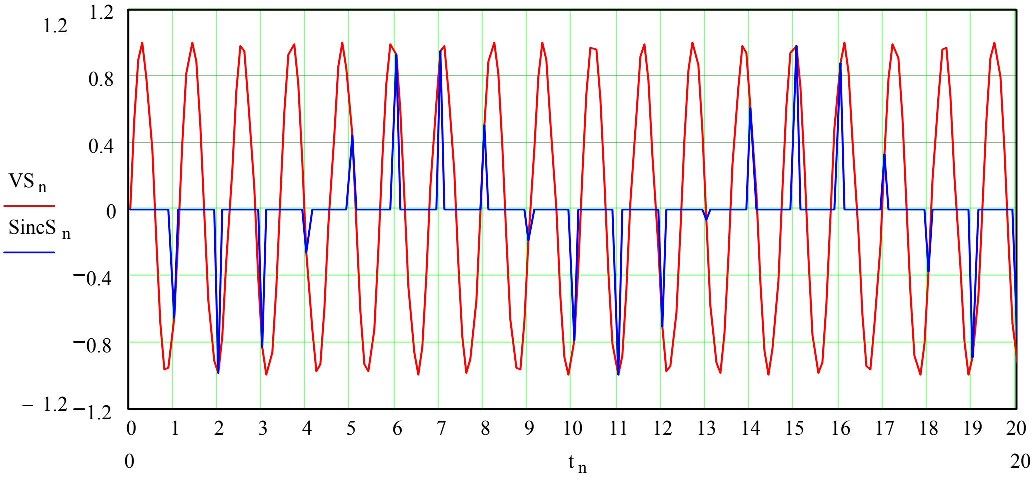
Copyright Dan Lavry, Lavry Engineering, Inc, 2004

The center of the wave (t=7 to t=45) is far enough from the sudden gating (t=0 and t=50). The center (say t=25) is impacted mostly by nearby sincs and little by far away sinc's The plot below shows the above wave with a black sinc at t=2 and blue sinc at 48. Note that their contribution around t=25 is about zero.

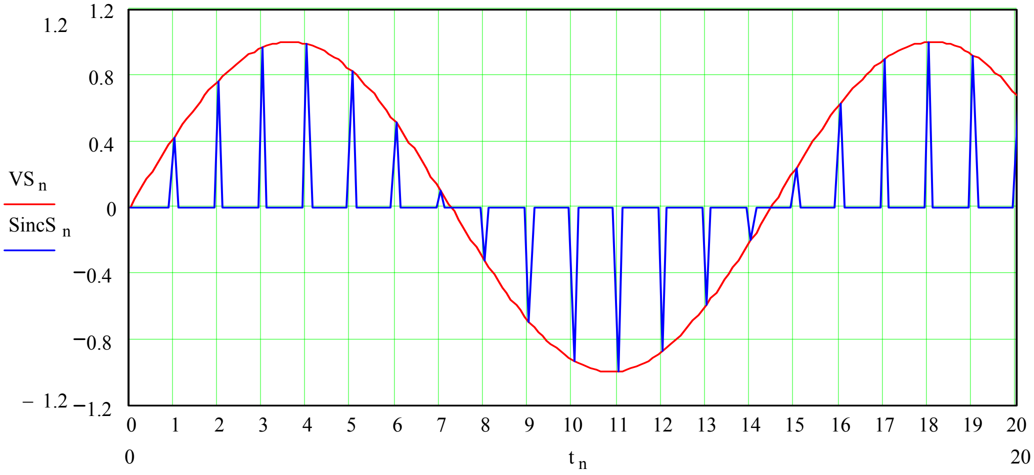


Each sample point is an outcome of all the sinc's, yet those sinc's nearby the sample point contribute more to the outcome. Our example consists of only 50 sinc's. A red book CD format is based on 44100 sinc's per second. Does it mean that the first few samples on a CD distorted? The answer is NO. The example above is made of a sine wave burst, and the reason for the distortions at the beginning and end is the high frequency energy content of the burst. Lowering the high frequency energy will reduce the distortions. In fact the distortions will disappear completely when the input signal has no high frequency energy.
At this point we need to clarify what we mean by "high frequency" signal: "high frequency" is any frequency above the "ringing" of the sinc function. The general shape of the sinc function is sinc = sine (X) / X . What is X? The specific sine wave used to construct the sinc function sets the limits for our process of being able to reconstruct a wave out of a sum of sincs. The sinc ringing frequency must be higher than twice that of the highest possible frequency we want to sample. Not meeting the requirements will cause distortions, called aliasing distortions.

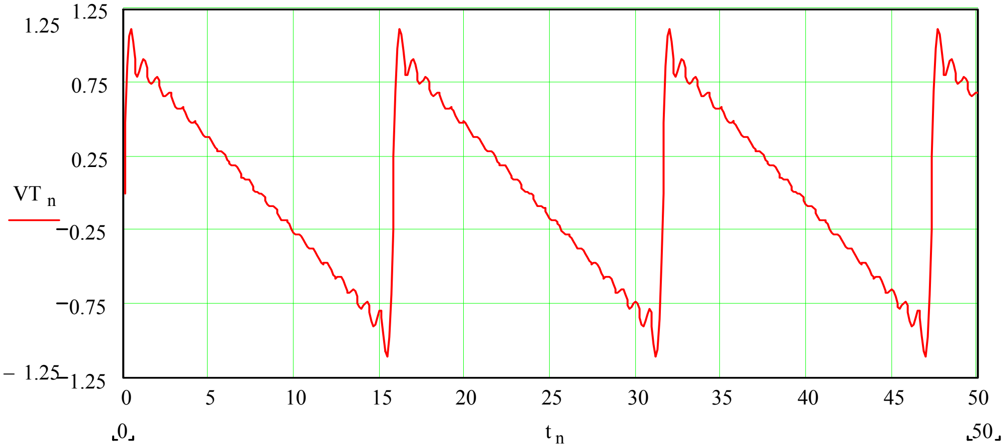
Let us examine sampling of a sine wave with frequency above half of the sampling rate. The red wave is the high frequency input. The blue shows the locations and magnitude of the sampling sinc's. Clearly in this case, we do not have enough sample points to track the high frequency. The sum of the sinc's will yield a wrong result. We must sample faster than twice the signal bandwidth.



In the previous plot, the occurrence of the sinc's is too slow to track the fast changing input wave. Let's slow the input wave to less than half the sampling frequency. The sinc's location is now properly set to track the waveform.



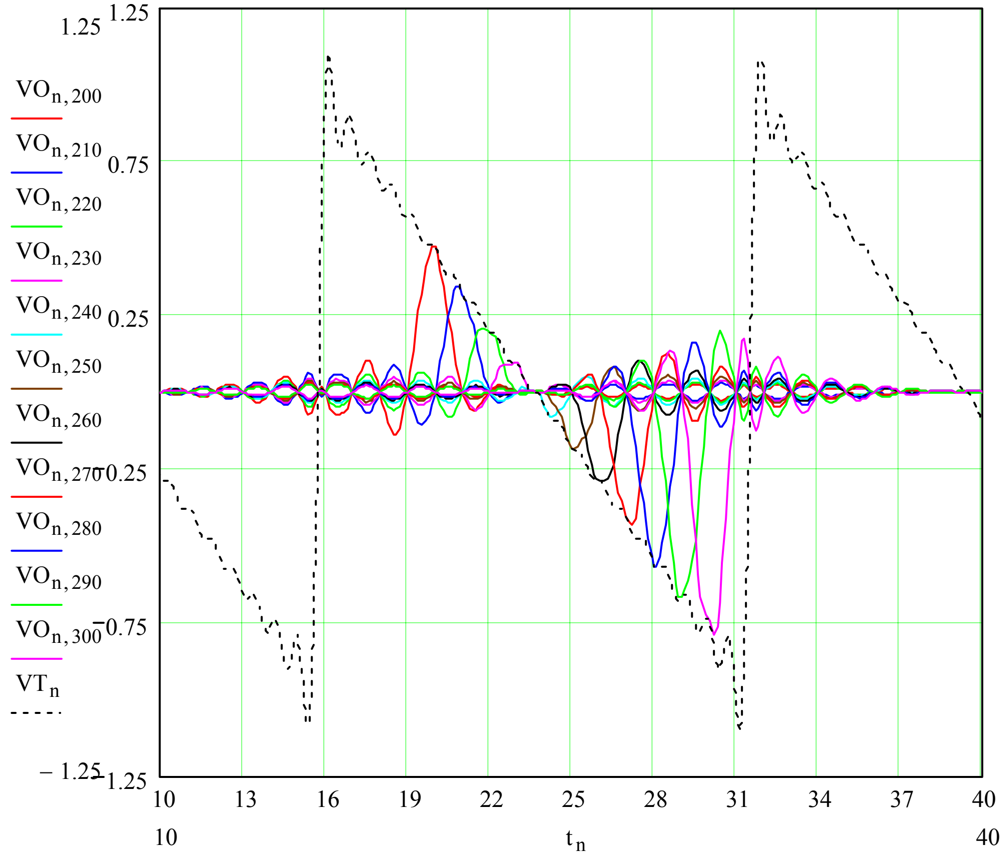
Let's investigate a more complex waveform. We can repeat the above process for a band limited "saw tooth" wave. Below is such a wave, made by adding 16 harmonics. All harmonics above 22KHz are set to zero.



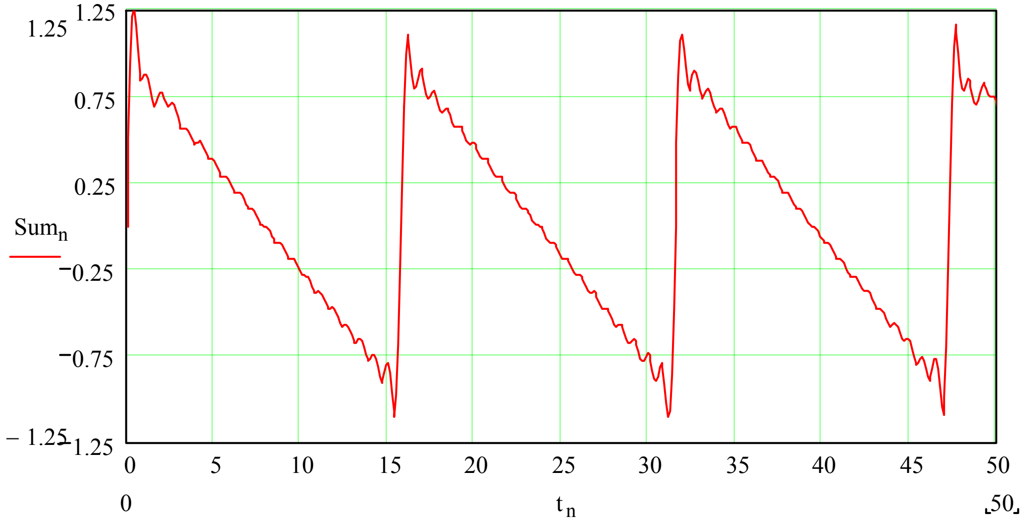
Both the ringing and the finite rise time are a result of band limiting. A perfect saw tooth would take infinite bandwidth.

Let’s align and multiply a few sinc's with the saw tooth wave:

Copyright Dan Lavry, Lavry Engineering, Inc, 2004

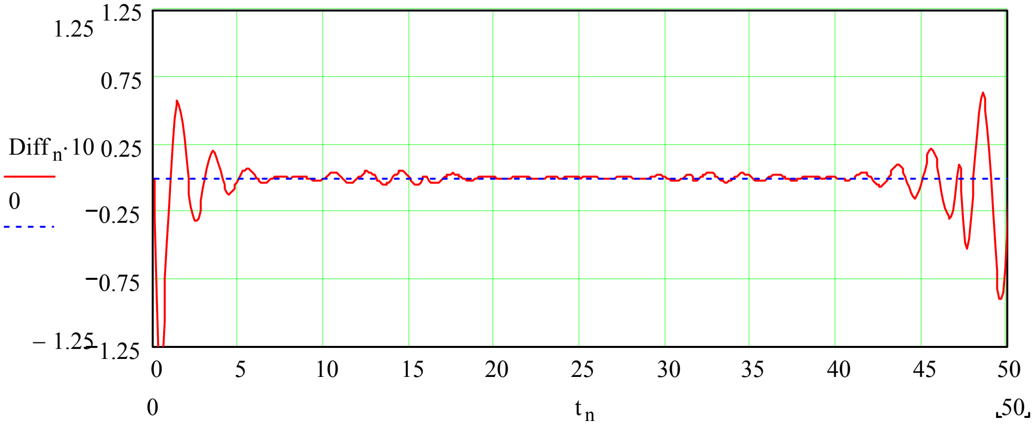


Let us complete the process by adding ALL the sincs functions. Again we see some distortions at the abrupt start (near t=0) and the end (t=50). The more abrupt, the bigger the distortion.



Let's find the difference between the original band limited wave and the sum of the sincs. The plot below shows the difference magnified by X10. Note that the error converges to zero at around t=25. The center point T=25 is only 25 sinc's away from the start and stop points.

Copyright Dan Lavry, Lavry Engineering, Inc, 2004



Once again, the errors near the ends (start and stop) are due to the high frequency content near the ends of the input signal. Keep in mind that the error is a high frequency signal.

Let us review Nyquist Sampling Theory: A sampled waveforms contains ALL the information without any distortions, when the sampling rate exceeds twice the highest frequency contained by the sampled waveform.

Note that once we agree on what constitutes audio bandwidth, we need not sample much faster than twice that bandwidth to have the ability to retrieve 100% of the original signal. How can we retrieve the signal from a series of sample points? We already know one way to do that. We align our sinc's with our sample times, scale them (multiply) and sum it all. That is nice in theory, but where can we find such sinc functions in hardware? Is there any additional meaning to the sinc function? So far we viewed the sinc as a time domain wave - a plot of amplitude variation in time. Is there another meaning to it?

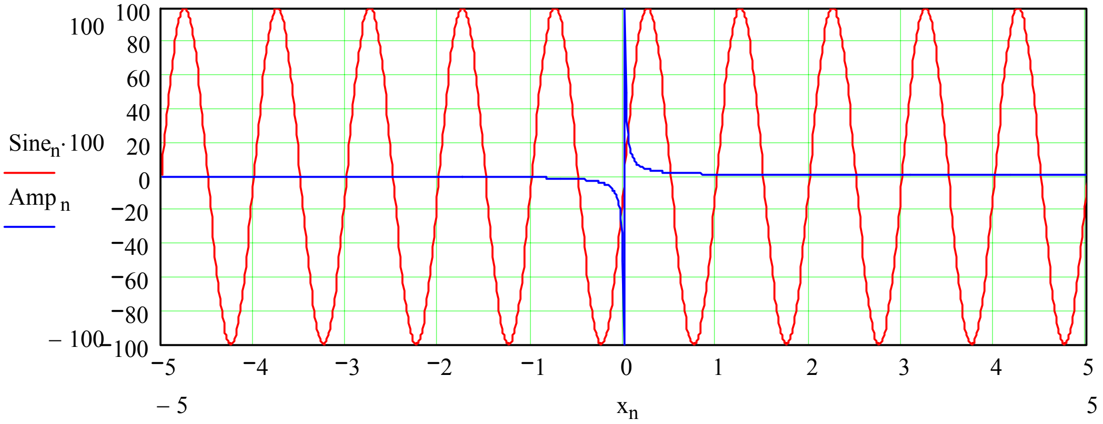
Time domain and Frequency domain

We saw how we can construct any band limited waveform out of sinc functions. We pointed out that there is no reason to sample at rates higher than twice the bandwidth, because the addition of the sincs yields the proper wave shape. The bandwidth limitation provides sufficient basis for perfect results (it is analogous to needing only 2 points to draw a straight line). However, we require that the frequency of the sinc function will exceed the highest frequency of the signal we need to sample. Such statement in the frequency domain can be restated in the time domain as: the sample width of the sinc (the distance between zero crossings) will not be longer than the cycle time of the highest frequency within the bandwidth. Next we will present some graphs and explain the relationship between sinc function and bandwidth.

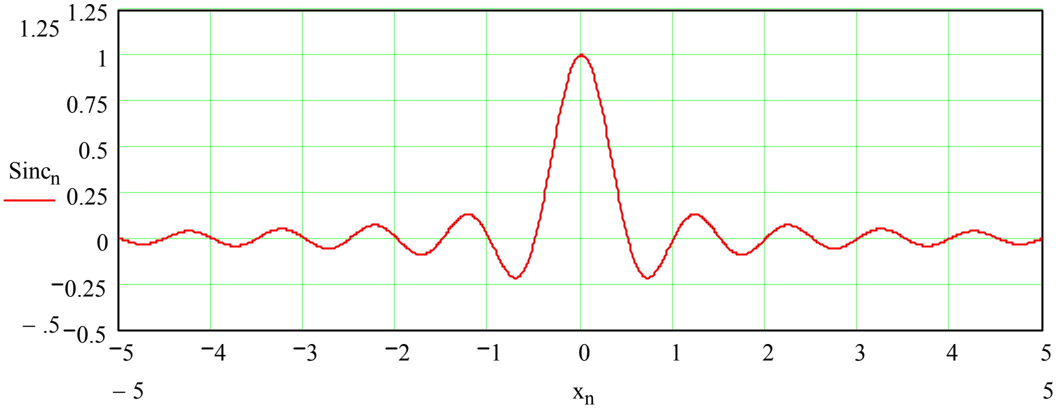
The traditional definition of sync is:

$\mathrm{sinc}(x) := \dfrac{\sin(x)}{x}$ for all x not equal 0, and $\mathrm{sinc}(x) := 1$ for x=C

Graphically speaking, we can see that sinc(x) is a product of sin(x) and 1/x. The plots below show a sine wave (red) and 1/X function (blue). The sine wave is amplified by X100 for the sake of clarity.



Multiplying the red and blue plots above yields a sinc function. Note that the positive half cycle near zero is multiplied by positive value and the negative half cycle near zero by a negative value. This generates a symmetrical outcome. The 1/x amplitude multiplier is 1, 1/2, 1/3, 1/4... at X=1, 2,3,4... respectively, making for a decaying envelope sine wave. The region near zero is less intuitive to understand, but we can see a multiplication of large envelope numbers (blue) by small sine wave near zero (red). In fact at x=0 where we encounter multiplying infinity by zero, we set the outcome to 1 to make a continuous and smooth plot at x=0. The outcome is the sinc function as shown bellow:



Authors comment: While I find the definition for a sinc function rather straight forward, but the function itself has amazed me over the years, specifically the "mysterious connection" between its ability to provide perfect interpolation of a band limited signal. I decided to try and understand it from a different angle. Let me begin by stating a theorem that will shed light on the relationship between the frequency domain behavior of a system and its time domain response. That is the relationship between bandwidth and the sinc function.

Lavry's sinc function theorem: The sum of all cosine waves of amplitude A within a bandwidth 0 to BW (Hz) is equal to a sinc function with BW frequency and BW times A amplitude.

Proof:

Copyright Dan Lavry, Lavry Engineering, Inc, 2004

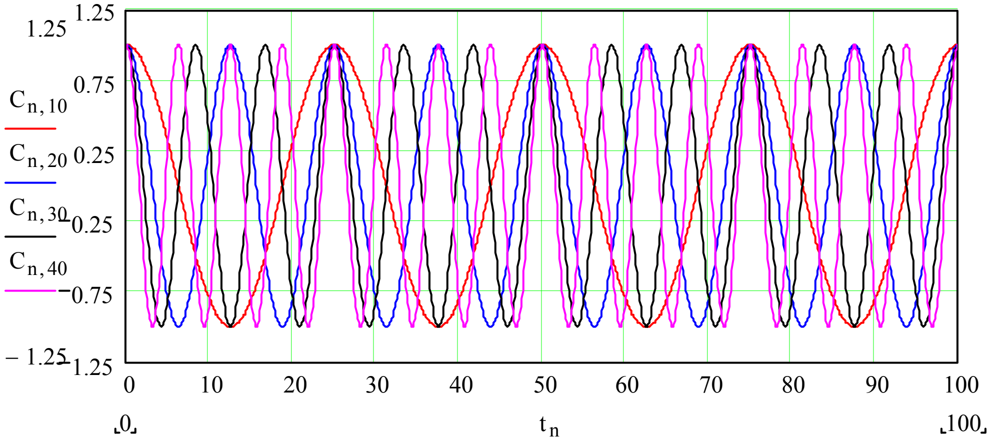
$$\int_0^{BW} A\cdot\cos(n\cdot t)\,dn \quad \text{is equal to} \quad (A\cdot BW)\cdot\frac{\sin(BW\cdot t)}{(BW\cdot t)}$$

Note: we use cosine waves (not sine waves) because the idea here is to take all the possible frequencies within the bandwidth and align the peaks of one cycle of each frequency at one point on the X axis. When using cos(x) waves, all the peaks occurs at x=0.

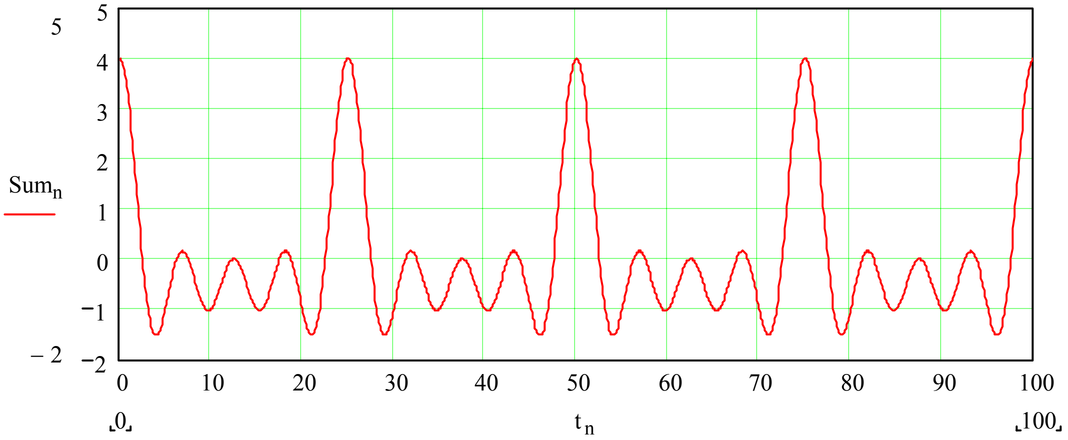
The mathematical proof is rather straight forward, yet the theorem points a direct relationship between bandwidth and the time domain impulse response.

A sinc wave in the time domain is an expression of a perfect brick wall low pass filter. Let us investigate this relationship graphically.

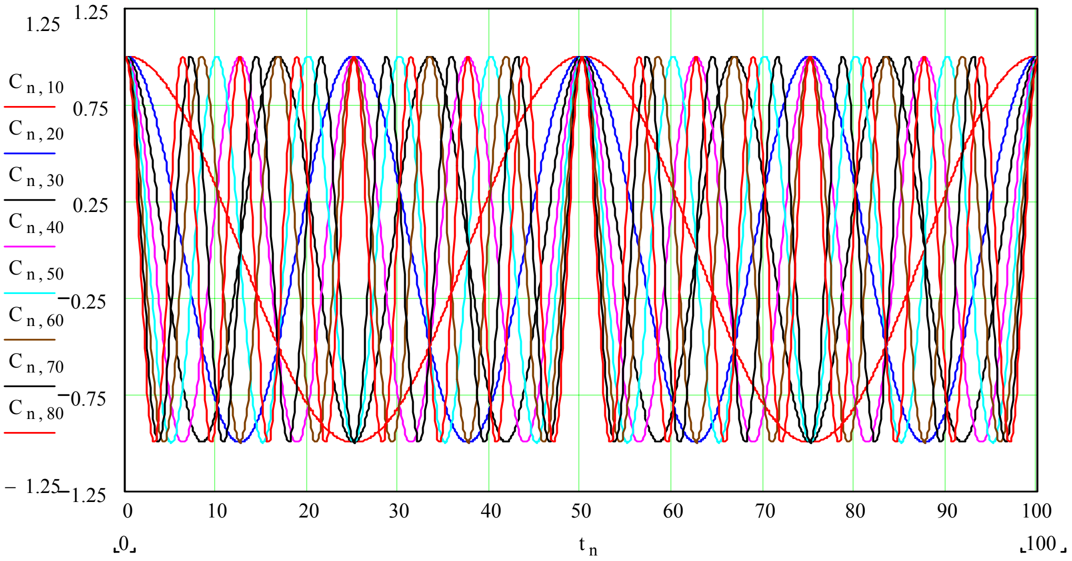
We start by plotting 4 cosine waves at the following frequencies:
1Hz, 2Hz, 3Hz and 4Hz.



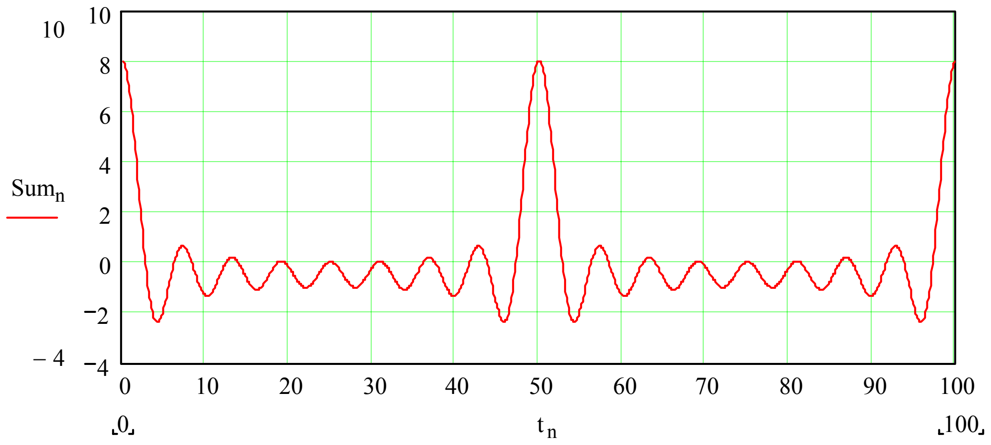
Adding the 4 waves yields and interesting result. We see the 4 waves reach their maximum value at t=0, t=25 and t=50, t=75 and t=100. This is where the added outcomes peak. Conversely, at say t=13, while two waves are peaked, the other two waves are at a minimum value, making the sum at t=13 zero. The sum of the 4 waves begins to look like a crude approximation of repeated sinc functions, such as we use for sampling:



Let us double the number of cosine wave. Our Bandwidth is still 4Hz, but we now have 8 waves separated by 1/2Hz. The frequencies are 1/2Hz, 1Hz, 1.5Hz, 2Hz, 2.5Hz....4Hz. Again, we use cosine wave (phase shifted sine waves) because we want the peaks to line up and add up to a "main lobe".



The sum of the above 8 waves yields the plot below.

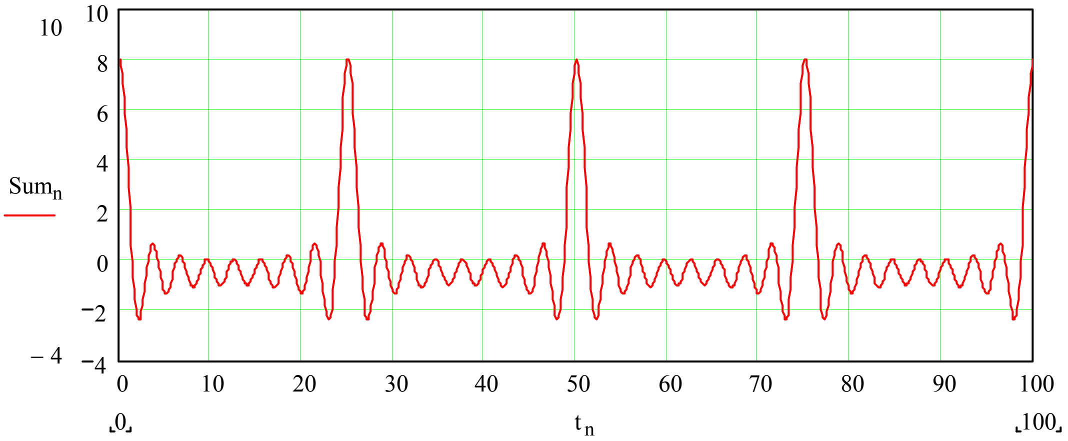


Let us analyze what happened. First, including more tones within our bandwidth makes a better approximation of a sinc. Note the decaying ringing. The more harmonics we add, the better the sinc approximation.

Second, we doubled the number of cosine waves and the result yields half the "approximate sinc's". With 4 waves we had 5 peaks (at t=0, 25,50,75 and 100). With 8 waves we get 3 peaks (at t=0, 50 and 100). The more waves we add, the further the separation between the peaks. Given that the bandwidth we chose 0-4Hz contains infinite frequencies, by the time we add them ALL, the separation will be infinite and we will end up with ONE sinc function.

Third, the width is a function of the bandwidth: increasing the bandwidth yields a narrower sinc. Conversely, the width of the sinc (the time between zero crossing of the “main lobe”) occurs at a frequency equal to the bandwidth of the system.

The next plot shows what happens when we increase the bandwidth by a factor of 2. The previous plot was made by summing 1/2Hz, 1Hz, 1.5Hz....4Hz. The plot below is made by summing 1Hz, 2Hz, 3Hz...8Hz. We doubled the bandwidth (from 4Hz to 8Hz). The overall shape is the same in both cases (8 waves) but the times scale is contracted - the sinc width is halved and there are twice as many sinc's The sum made of higher frequencies created narrower sinc approximation, because we are adding narrower half cycles with the "main lobe". We also see that the number of peaks doubled, but we already saw how to decrease them: all we need to do is insert more cosine waves. At the limit, when we have a cosine wave at every frequency over the selected bandwidth (there are infinite such frequencies) the sum becomes a perfect sinc, and there is only one "main lobe". That limit is an expression of analog brick wall filter - constant frequency to the cutoff frequency, zero amplitude above the cutoff frequency



Counter to the Fourier series that is based on a fundamental frequency and its integer multiples; the approach we have taken is to include all the possible sine (or cosine) waves. In the Fourier series case, we use sums of **in phase** sine and cosine waves to describe a periodic function. Our analysis is based on infinite number of waves of all frequencies, thus there is no common denominator to define a repetitive cyclical behavior. We can expect "only one cycle". In our case it is a "one cycle sinc function".

Note that while Fourier series is limited to the construction of **periodic** waveforms, the sinc function we constructed out of infinite waves is not subjected to requirements of periodicity.

With time limits set between -infinity and +infinity, we can expect one and only one point in time where all our waves will peak simultaneously at the center of our sinc. Although more complicated sums of waveforms could result in a sinc function, a set of cosines is the most convenient, because they constructively add at only one point : at time equal zero.

We are now equipped with an intuitive understanding of the relationship between a sinc function and bandwidth. Note that the outcome of increasing the bandwidth is a narrower sinc, and at the limit (infinite bandwidth), the sinc will become a theoretical impulse (zero width vertical line).

Practical vs. theoretical filters

While most of the energy of the mathematically pure sinc function is concentrated around the center (main lobe), the decaying ringing continue in both directions for ever. The amplitude gets smaller as we get further from the center. From a practical standpoint (such as meeting certain performance accuracy) we could simply ignore what happens far enough from center. In fact, when designing sinc based digital signal processing gear (such as FIR filters) we can not have a sinc waveform that lasts for ever. There are methods yielding excellent approximation to a theoretical sinc function. These short duration sinc like waveforms provide near theoretical results. This is covered in most Digital Signal Processing introductory literature (under the subject of FIR filters).

There are two other filters used in conjunction with sampling, reconstruction and general signal processing:

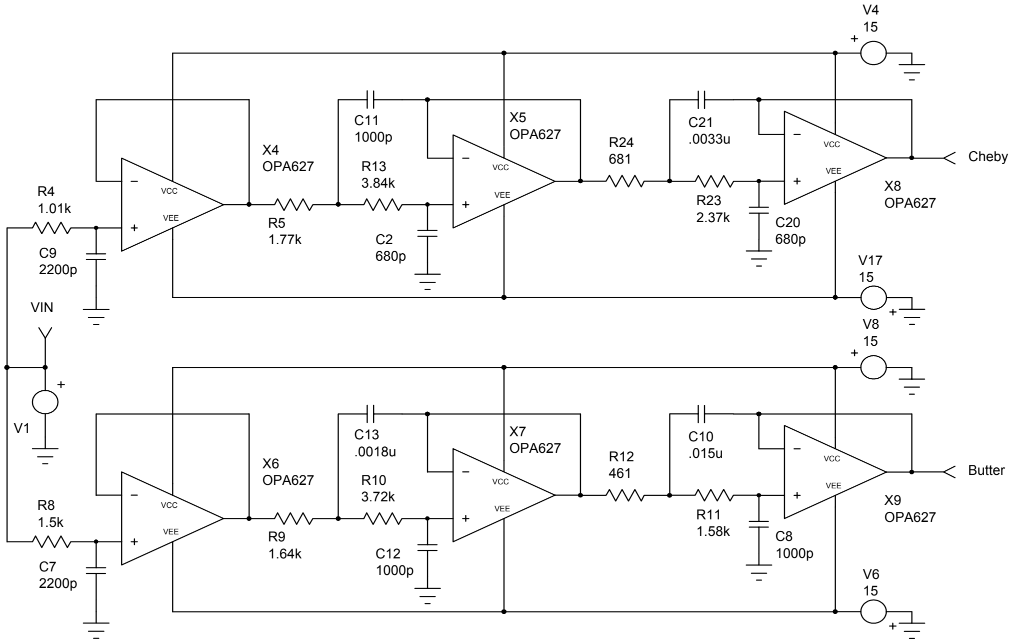
1. Analog filters. We use these “analog in, analog out” circuits as anti alias filters (before AD), anti imaging (after DA).
2. Infinite impulse response (IIR) filters. They are the near equivalent of an analog filter.

These filters are “digital in, digital out”

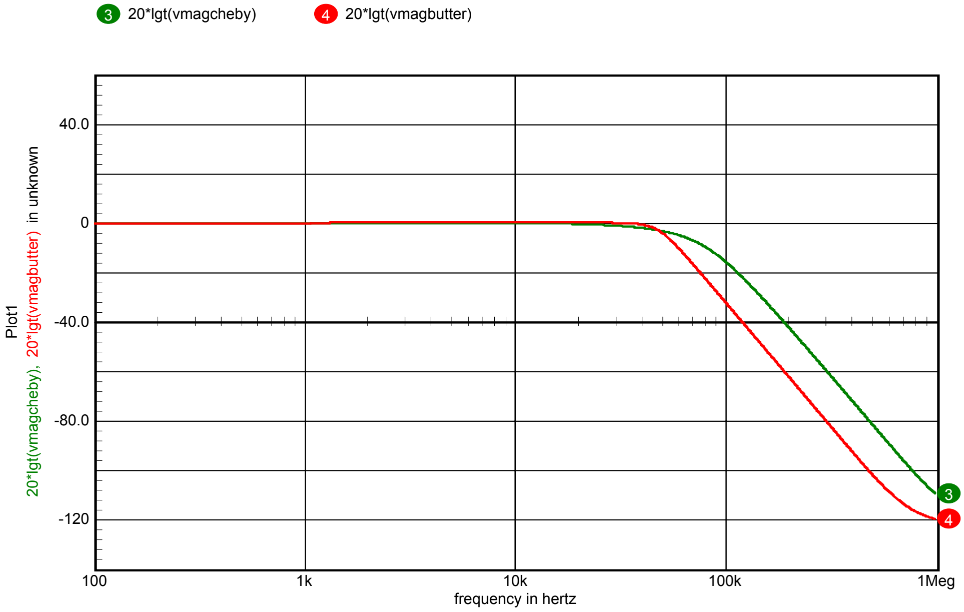
The theory presented so far showed signal reconstruction as a “sum of sinc functions”. The concept works because we chose a sinc function that is a filter with a bandwidth at Nyquist (half the sample rate). Nyquist did not guide us to use a sinc function based filter. He stated that the difference between the original wave and the original waveform is high frequency, thus the key to making them the same (to remove the difference) is filtering out high frequencies (above Nyquist).

Any filtering (removal) of the high frequency energy would remove the difference, thus yield the original. However we need to be sure that while filtering of the high frequencies we leave all the energy bellow Nyquist (the audio band).

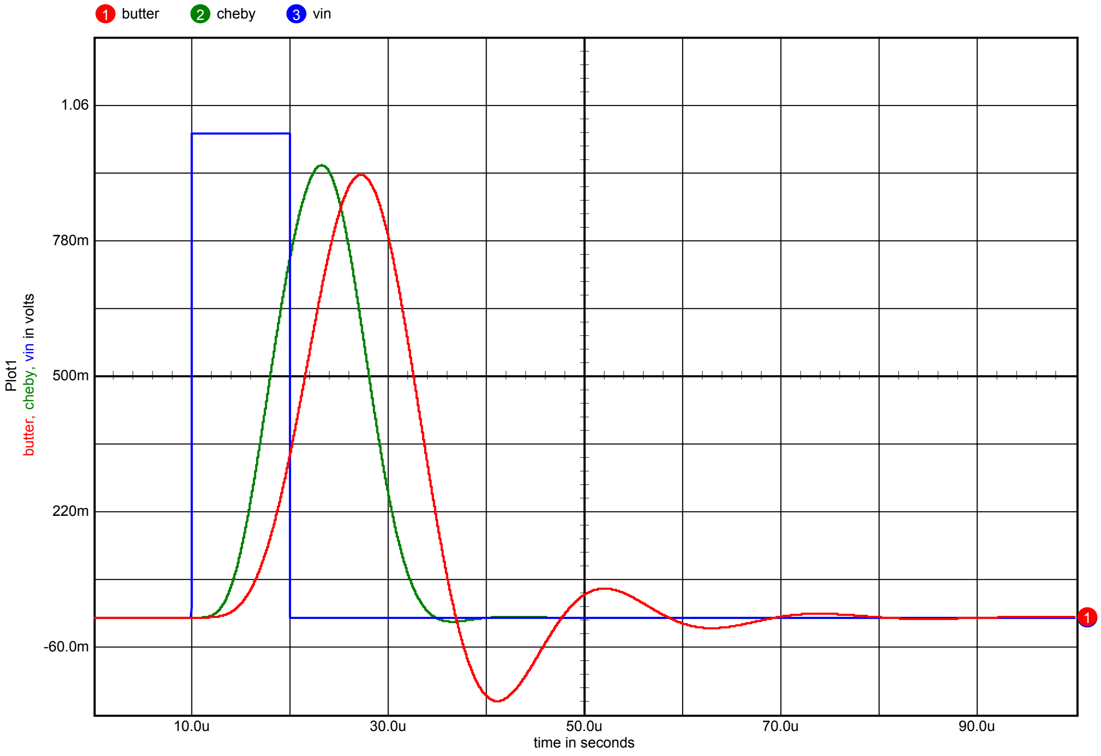
Bellow is a schematic of 2 analog filters, which may be used for anti aliasing or anti imaging. The input V1 (left side) drives both the top circuit (Chebyshev filter) and the bottom filter (Butterworth). Both circuits are 5 pole filters, with a bandwidth of 48KHz (-3dB point). The different resistor and capacitor values yield different responses.



Let us make the input V1 into a sine wave and sweep over 0 to 1MHz and plot the frequency response (red - Butterworth, green – Chebyshev):



Let us now look into the impulse response. Theoretically, we should make V1 into zero width impulse. Of course, if the time duration is zero, there is no energy, unless the amplitude is infinite. We will approximate the impulse by making V1 into a 10 uSec pulse. The plot shows the impulse in blue (1 volt between 10 and 20uSec, zero elsewhere). The two other traces show the circuit impulse responses (red - Butterworth, green – Chebyshev):



These are “real world circuits”. Unlike the theoretical sinc analysis, any activity at the output can not happen before the occurrence of the input.

It is interesting to see that the Butterworth filter (red) looks similar to a sinc function on the right side of the peak. Note that one cycle of ringing is about 21uSec (starting at 48uSec and ending at 69uSec). A 21uSec cycle time is 48KHz which is the filter bandwidth. The Butterworth is a “near relative” of the old sinc function; at least on the decay side.

The Chebyshev filter (green) yields a different impulse response. The impulse is slightly steeper, sooner and free of ringing. It is certainly does not look like a sync function.

Both filters provide examples of real world circuits that can be used for anti aliasing or anti imaging. It is removal of high frequency energy above Nyquist that counts (while keeping the audio in tact). A good filter yields the original waveform out of a sampled signal.

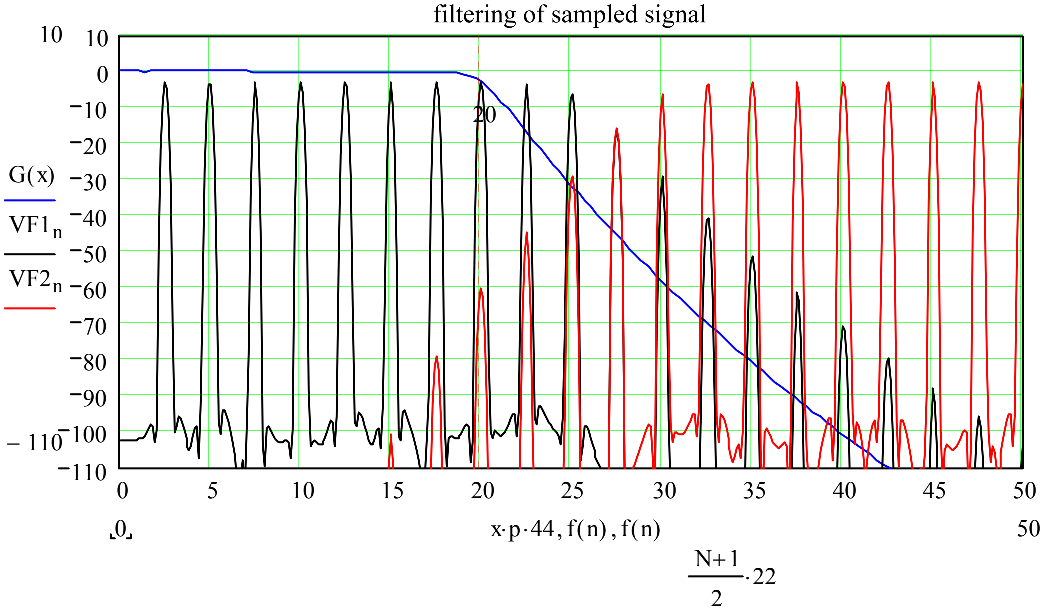
Other practical considerations

Let us now focus on some practical issues relating to sampling and reconstruction hardware. An Ideal AD (for sampling) followed by an ideal DA (reconstruction) would behave as well as a perfect wire, yielding a waveform out of the DA that is the same as the waveform fed into the AD (we can allow for gain or attenuation).
Retaining the **correct waveform** is the **scientific** criteria for perfect conversion. Many factors may cause deviation from the ideal conversion. Some of such deviations are intentional (EQ, reverb and more); others are undesirable and are caused by equipment shortcomings.

Filtering to avoid aliasing

To meet Nyquist's criteria, one needs to ensure that no energy at frequencies above half of the sample rate enters the AD converter. High frequency energy may find its way into the audio range (aliasing). Pre filtering may be needed to avoid such unwanted outcome. Early audio converters designers faced a serious compromise between the steepness of a hardware filter and the size of data files. High filter orders yield better alias rejection but increase circuit complexity and can increase noise and distortions.

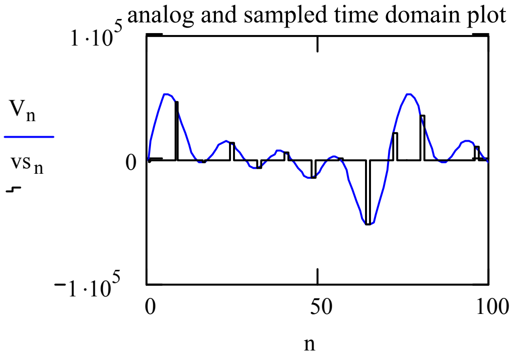
Blue: filter curve
Black: filtered audio
Red: filtered aliased energy

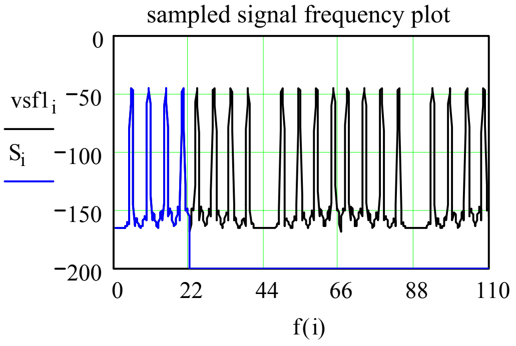


Removing high frequency BEFORE sampling (above Nyquist) is needed to avoid aliasing. We will next see that sampling itself introduces high frequency energy. The high frequency energy is “carried along” with the digital signal, but can be removed later.

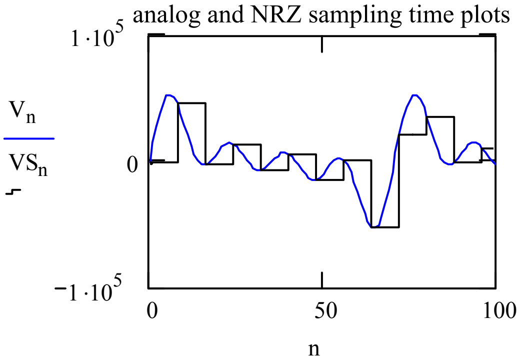
Sampling and the Sinc Function

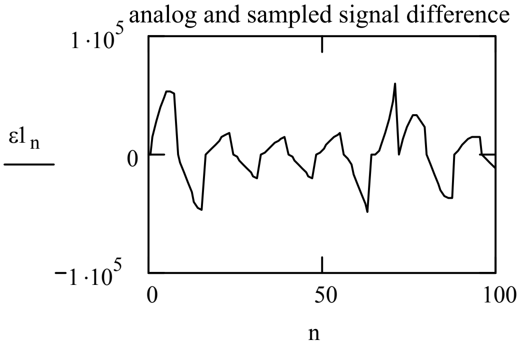
"Near theoretical" sampling, is a good starting point but not really practical. Zero width pulses carry no energy. Real hardware DA reconstruction require some minimum (non zero) pulse width. The signal shown below is made of 4 tones (time domain plot on the left and frequency plot on the right). The high frequency energy is part of the sampled signal.



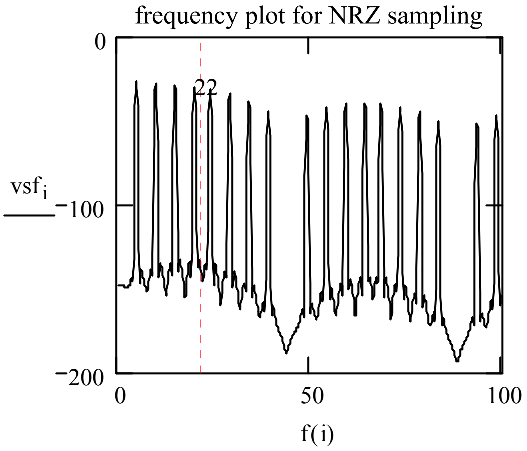


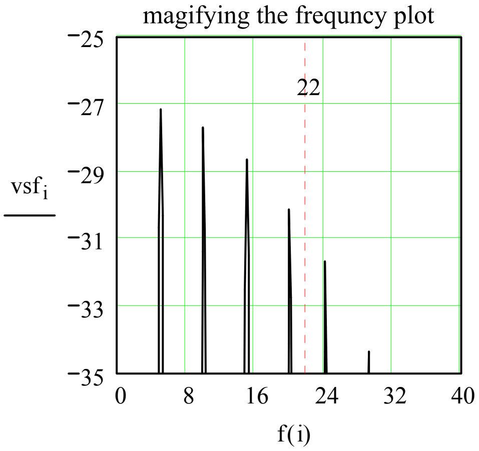
Most practical circuits extend the width of each pulse so the time of the next sample. It is called NRZ (not return to zero). The advantage of the wider NRZ pulse is higher energy. The sample value is no longer a point. It is retained (held) until the next sample time. The difference between the input signal and its NRZ representation is an error signal (right plot)





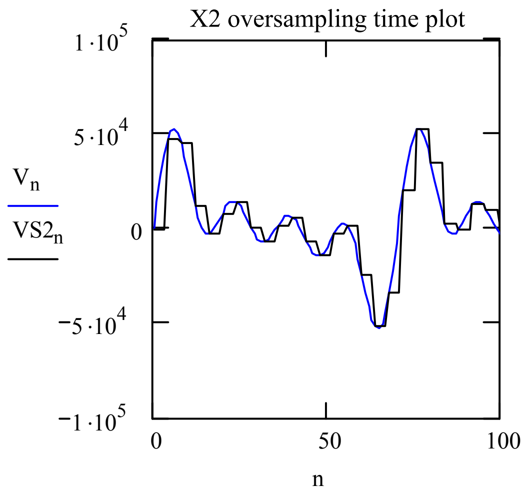
Most of the energy in the error signal is made of frequencies above the audio band, but some of that energy causes attenuation at audible high frequencies.
This attenuation can be observed on a frequency response plot (amplitude vs. frequency). There is no harmonic distortion associated with that attenuation, yet it is an undesirable side effect of NRZ sampling.

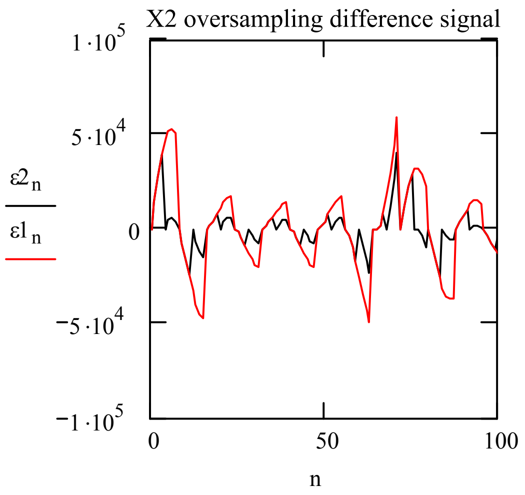




Over sampling (shown below X2):

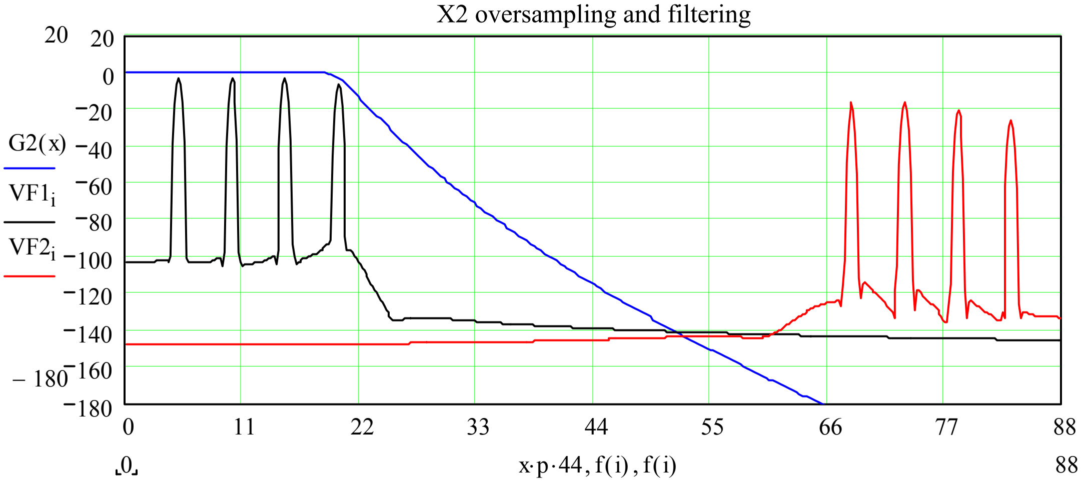
Sampling at a faster rate lowers the amplitude of the error (difference) energy. It also shifts the frequency content of the error signal to higher frequencies. The error (right plot) shows X2 in black (X1 in red)





With X2 sampling, both AD alias protection requirement and DA high frequencies image removal are located at much higher frequencies thus pre and post filtering is much easier. The undesirable high frequency attenuation due to NRZ sampling has been reduced as well

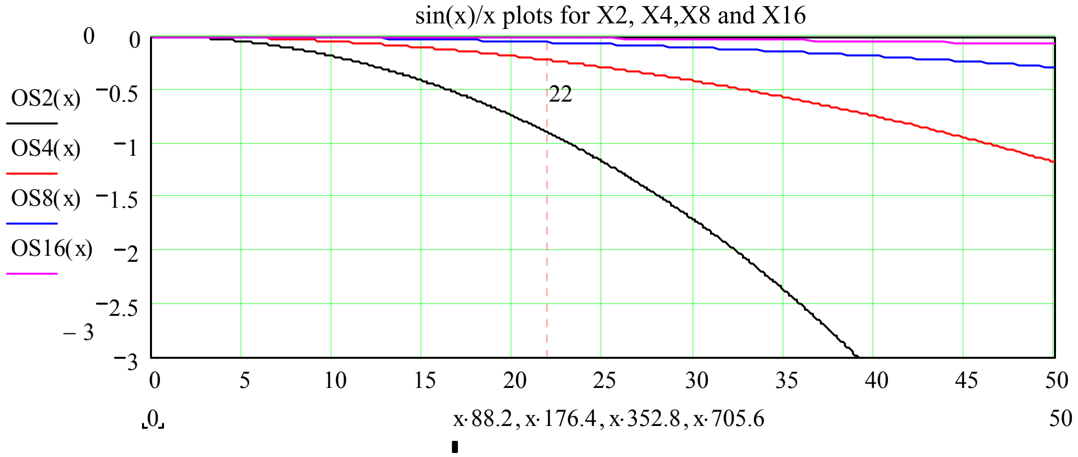
Blue: filter curve
Black: filtered audio
Red: filtered aliased energy



Note the additional advantage of over sampling: the attenuation of high frequency audio is reduced. The faster we over sample the less high frequency audio band attenuation. The plot below shows the improvement in flatness response for increased over sampling ratios. As mentioned before, this attenuation is due to NRZ sampling. Notice the attenuation near 20KHz:

Black: X2 over sampling
Red: X4 over sampling
Blue: X8 over sampling

Purple: X16 overselling



Clearly there are benefits to faster sampling:
1. Easier filtering (AD anti aliasing and DA anti imaging)
2. Reduction of higher frequencies attenuation at the DA side.

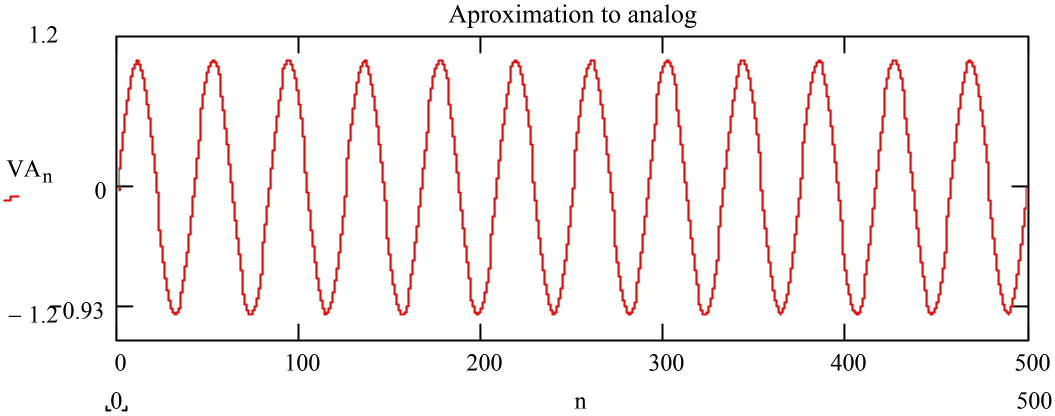
Indeed, such faster sampling is common practice with both AD and DA hardware. Most AD's today are made of two sections: a front end (modulator) and a back end (decimator). The front end operates at very fast rates (typically at 64 -512 times faster then the data output rate). The reasons for such fast operation is outside the scope of this article. It is sufficient to state that anti alias filtering and flatness response becomes a non issue at those rates.

It is most important to avoid confusion between the modulator rate and the conversion rate. Sample rate is the data rate. In the case of AD conversion, the fast modulator rate (typically less bits) is slowed down (decimated) to lower speed higher bit data. In the case of DA converters, the data is interpolated to higher rates which help filtering and response. Such over sampling and up sampling are local processes and tradeoff aimed at optimizing the conversion hardware.

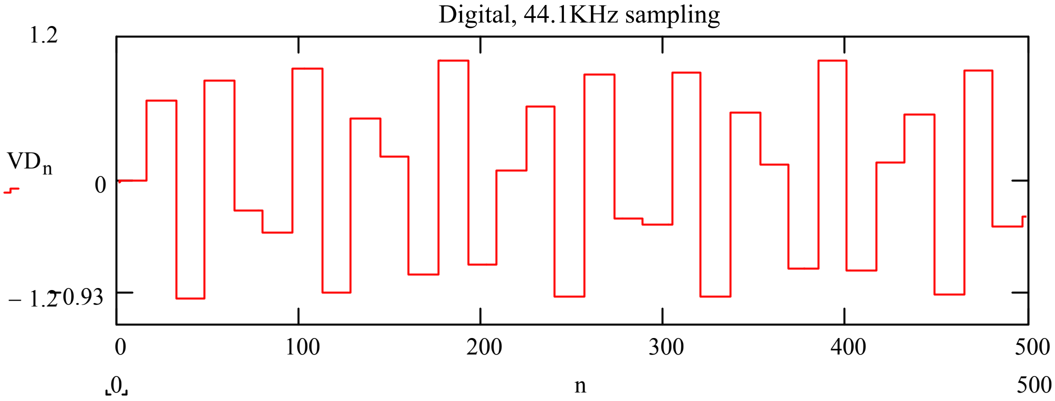
One should not confuse modulator speed or up sampling DA with sample rate, such as in the case of 192KHz for audio.

You do not need "more dots" for better accuracy. More examples and plots:

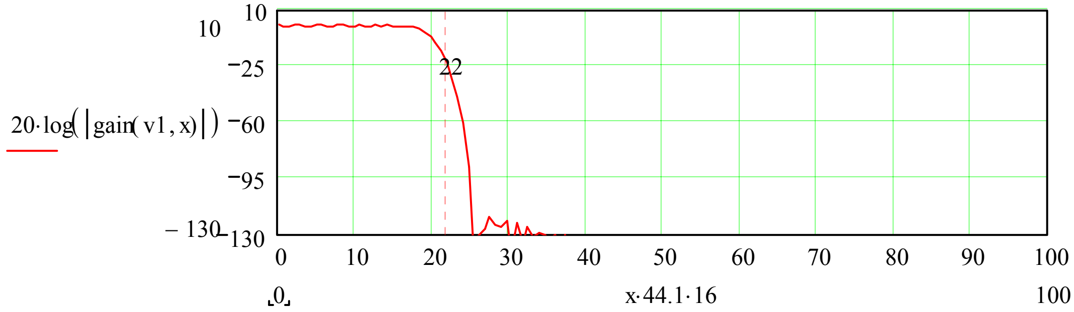
How many sample points are needed? Analog signal is continuous thus it has infinite points. Therefore analog simulation is not computer friendly. For the sake of demonstrating our case, let us approximate (approach) an analog signal by sampling the audio at 705.6KHz (44.1KHz X 16). The tone frequency is 17KHz.

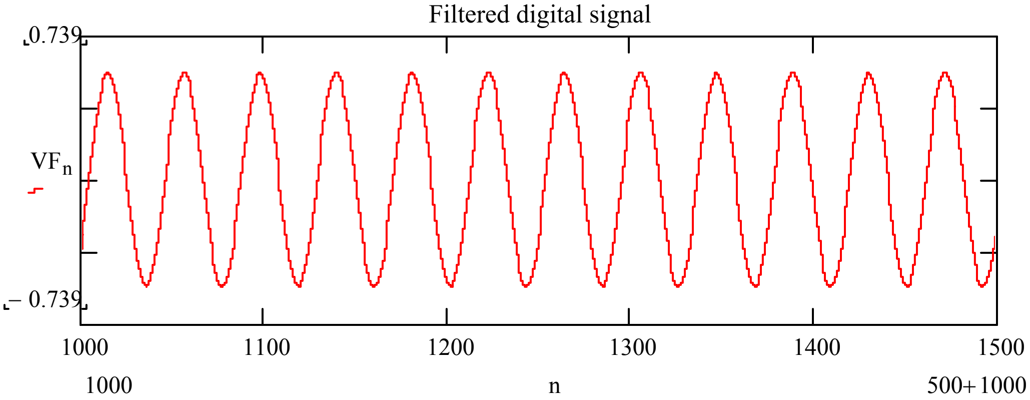


The 17KHz sine wave does not look like much of a sine wave:



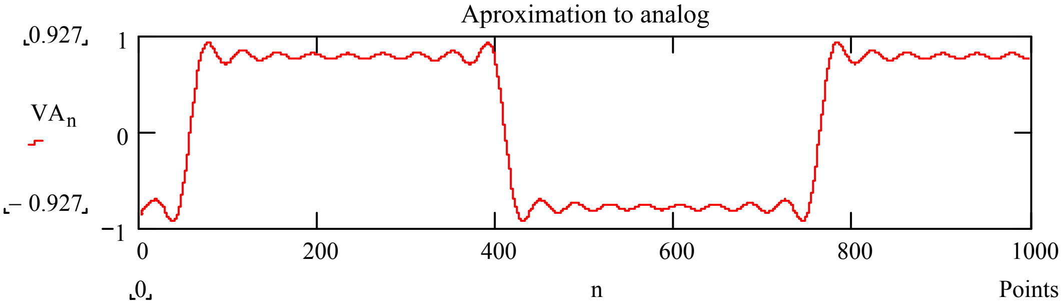
Let us filter the above digitized (sampled) signal. We will again approximate an analog output with an X16 digital approximation. Below are filter response plot (0-100KHz) and the output waveform, which again looks like a 17KHz "analog tone"

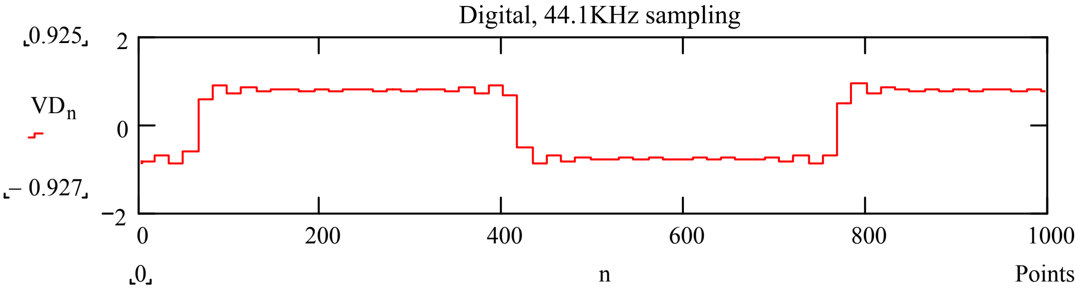




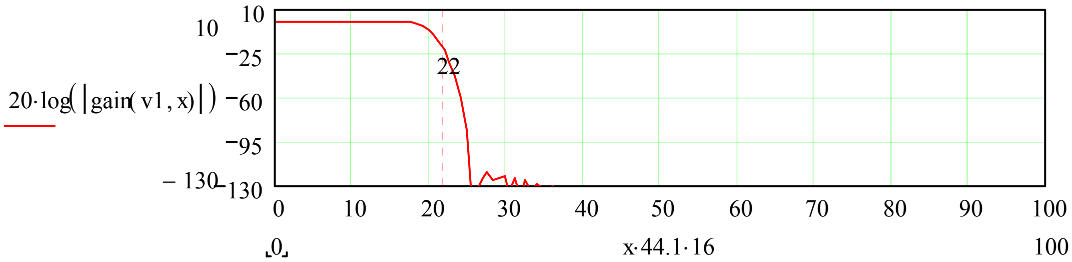
Let’s repeat the process for a 1 KHz square wave

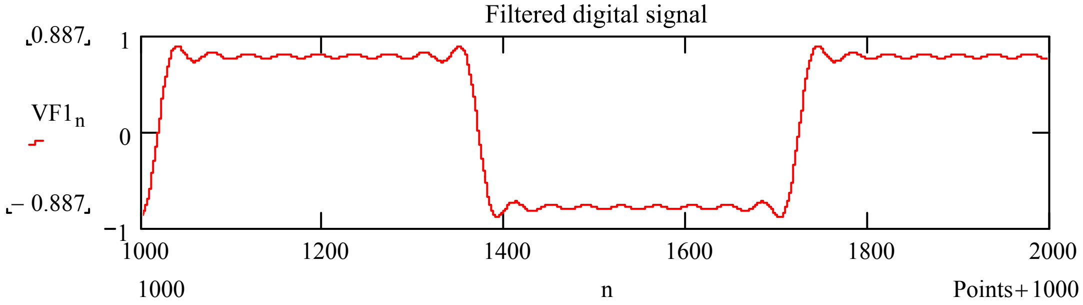
The wave is not "perfectly square" because it is bandwidth limited to 20KHz. This is what an audio square wave looks like.





Let us filter the digitized signal. We will again approximate an analog filter with an X16 digital approximation.





We saw 2 examples of sampling (AD) followed by reconstruction (DA).
It is important to realize that the end result yields a waveform where the values are correct, not just at sample times but at all times.
You DO NOT need more dots. There is NO ADDITIONAL INFORMATION in higher sampling rates. As pointed out by the VERY FUNDUMENTAL Nyquist theory, we need to sample at above twice the audio bandwidth to contain ALL the information.

So we have the pros and cons to increased sampling:

Pro: Easier filter
Overcome Sinc problem

Con: Reduced accuracy
Significant increase in data files size
Significant increase in processing power required

We can optimize conversion by taking advantage of concepts such as over sampling, up sampling and decimation. These processes help the hardware at the proper locations (AD and DA) and should not be confused with system sample rate. The determination of sample rate must be decided by **bandwidth** of the ear.

Record at 192KHz than process down to 44.1KHz?

There are reports of better sound with higher sampling rates. No doubt, the folks that like the "sound of a 192KHz" converter hear something. Clearly it has nothing to do with more bandwidth: the instruments make next to no 96KHz sound, the microphones don't respond to it, the speakers don't produce it, and the ear can not hear it.

Moreover, we hear some reports about "some of that special quality captured by that 192KHz is retained when down sampling to 44.1KHz. Such reports neglect the fact that a 44.1KHz sampled material can not contain above 22.05KHz of audio.

Some claim that that 192K is closer to the audio tape. That same tape that typically contains "only" 20KHz of audio gets converted to digital by a 192K AD, than stripped out of all possible content above 22KHz (down sample to CD).

“If you hear it, there is something there” is an artistic statement. If you like it and want to use it, go ahead. But whatever you hear is not due to energy above audio. All is contained within the "lower band". It could be certain type of distortions that sound good to you. Can it be that someone made a real good 192KHz device, and even after down sampling it has fewer distortions? Not likely. The same converter architecture can be optimized for slower rates and with more time to process it should be more accurate (less distortions).

The danger here is that people who hear something they like may associate better sound with faster sampling, wider bandwidth, and higher accuracy. This indirectly implies that lower rates are inferior. Whatever one hears on a 192KHz system can be introduced into a 96KHz system, and much of it into lower sampling rates. That includes any distortions associated with 192KHz gear, much of which is due to insufficient time to achieve the level of accuracy of slower sampling.

Conclusion:

There is an inescapable tradeoff between faster sampling on one hand and a loss of accuracy, increased data size and much additional processing requirement on the other hand.

AD converter designers can not generate 20 bits at MHz speeds, yet they often utilize a circuit yielding a few bits at MHz speeds as a step towards making many bits at lower speeds.

The compromise between speed and accuracy is a permanent engineering and scientific reality.

Sampling audio signals at 192KHz is about 3 times faster than the optimal rate.
It compromises the accuracy which ends up as audio distortions.

While there is no up side to operation at excessive speeds, there are further disadvantages:

1. The increased speed causes larger amount of data (impacting data storage and data transmission speed requirements).
2. Operating at 192KHz causes a very significant increase in the required processing power, resulting in very costly gear and/or further compromise in audio quality.

The optimal sample rate should be largely based on the required signal bandwidth. Audio industry salesman have been promoting faster than optimal rates. The promotion of such ideas is based on the fallacy that faster rates yield more accuracy and/or more detail. Weather motivated by profit or ignorance, the promoters, leading the industry in the wrong direction, are stating the opposite of what is true.