Audio Engineering Society
Convention Express Paper 109
Presented at the 154th Convention
2023 May 13–15, Espoo, Helsinki, Finland

Is oversampling always the solution?
Alternative strategies to minimize aliasing in dynamic processors

Juan Sierra[1]

[1] *NYU*

Correspondence should be addressed to Juan Sierra (`jds833@nyu.edu`)

ABSTRACT
Aliasing is an inherent condition of discrete signals, which can occur when manipulating the signal, even after the process of digitizing analog signals. In dynamic processors like compressors, spectral expansion produced by signal manipulation can lead to aliasing. Traditionally, oversampling has been the most common approach to deal with this problem. This means increasing the sample rate of the signal to allow spectral expansion before returning to the original sample rate once proper band-limiting has been applied. However, another solution is to band-limit the involved signals prior to compression. In this paper, we compare the effectiveness of these strategies and quantify them. Our comparison highlights the importance and power of measuring aliasing and demonstrates how to perform a *black-box* measurement of dynamic processors, as long as they have a side-chain input.

Understanding Aliasing

Aliasing occurs when a signal is sampled at a rate that is too low to capture its true frequency content. According to Oppenheim and Shchafer, this is because higher-frequency components of the signal are incorrectly interpreted as lower-frequency components due to insufficient sampling [1]. While aliasing often occurs during the digitization of analog signals, it can also occur through the processing of digital signals. This is because aliasing is an inherent property of discrete signals, as a finite set of discrete amplitudes can represent an infinite number of band-limited waveforms at different pass-bands [2].

Aliasing can occur when certain processes alter the spectrum of a signal in a way that is no longer band-limited by the Nyquist frequency. In such cases, the spectrum that surpasses the Nyquist limit is represented by a set of points that can also represent a different in-band spectrum. As a result, the aliased spectrum combines with the original band-limited spectrum, making them indistinguishable from one another. This is because the output of the process now contains aliased components, and we can no longer separate the in-band spectrum from the out-of-band aliases.

One simple example of aliasing occurs when a memory-less non-linearity is implemented. The non-linear nature of the process expands the spectrum of

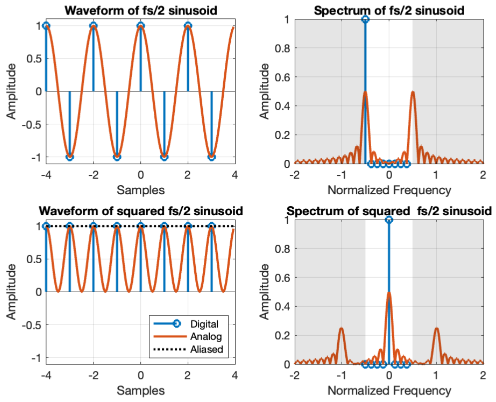

Fig. 1: Squared sinusoid at Nyquist frequency showing the effects of aliasing

the original signal, which may or may not produce aliasing. *Fig. 1* shows how a single sinusoid, when processed with a squaring non-linearity, produces a spectrum with new spectral content above the original one. This can be shown analytically through simple trigonometric identities, as demonstrated in *Eq. 1*.

$$y(t) = x(t)^2$$
$$x(t) = cos(\theta)$$
$$\cos^2(\theta) = \frac{1+\cos(2\theta)}{2} \tag{1}$$

In particular, if the original sinusoid represents the Nyquist frequency, squaring it will alias into a DC signal (note that since the digital signal is $[1,-1,1,-1,...]$, squaring it produces $[1,1,1,1,...]$). After the processing, we can no longer separate the produced spectrum into what should be preserved and what should be discarded due to corresponding to spectral content above the Nyquist frequency. As a result, aliasing has occurred, and it is not possible to distinguish between the in-band spectrum and the aliases of out-of-band spectrum.

Some non-linear processors, including some dynamic processors, take into account that the spectrum might be modified in such way that aliases are produced. The common approach is to use an oversampling scheme to allow the correct expansion of the signal's spectrum and let the aliases fall well beyond the original Nyquist limit [3]. When returning to the base rate, an anti-aliasing filter does its best to remove any frequencies above the original Nyquist limit [3].

Mapes-Riordan studied the effects of aliasing in dynamic processors from a worst-case perspective stating that to completely avoid aliasing an oversampling up to 5MHz would be needed [4]. Foti proposed and showed that aliasing was, to a great extent, a significant culprit behind the perceptual differences between analog and digital dynamic processors [5]. However, in both cases the objective measurements of aliasing were done based on spectral data that did not quantify the overall effect of aliasing; moreover, their analysis were based on sinusoids and analytical signals rather than complex waveforms.

However, in 1999 Thornburg proposed the concept of Aliasing Signal-To-Noise Ratio (ASNR) treating aliased spectrum as noise and non-aliased spectrum as signal [6]. Although this was to determine the amount of oversampling to be used in different non-linear models, it is possible to extend this idea and use it in dynamic processors. Moreover, it is possible to compare different aliasing reduction strategies and evaluate their relative effectiveness. Because this approach is often used only in analytical models it is widely implemented; however, a method to use this in practical situations, together with why it is important to rescue this measurement, will be later presented.

Dynamic Processor Structure

Fig. 2 presents a simple scheme for a dynamic range processor. The input signal, $x[n]$, is split into two paths: an unmodified path and a side-chain section. The side-chain implements an algorithm that produces a control signal, $g[n]$, which adjusts the amplitude of the input signal using simple multiplication to produce the output signal, $y[n]$ [7].

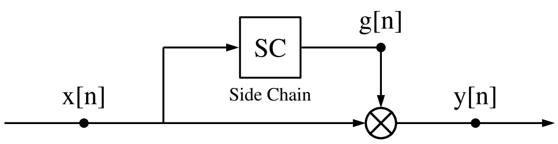

Fig. 2: Simple Feed-forward Dynamic Processor structure.

According to the convolution theorem, time-domain convolution is equivalent to the product of spectra. Similarly, the symmetry of Fourier theory implies that a time-domain product is equivalent to a convolution of spectra [8]. Thus, modifying the dynamic behavior of the input signal by multiplying it with a control signal is equivalent to convolving their corresponding spectra. However, this would lead to an increased bandwidth that is not accommodated by the element-wise product of time-domain signals. Instead, the product is represented by cyclic convolution, which introduces aliasing [9]. The aliasing produced is a result of the spectral convolution spanning more than the Nyquist bandwidth, which causes the spectral images to alias, resulting in circular convolution. *Fig. 3* visually compares cyclic and acyclic convolution of two vectors. If we could compute the spectral convolution without the images, the result would extend beyond Nyquist and be alias free.

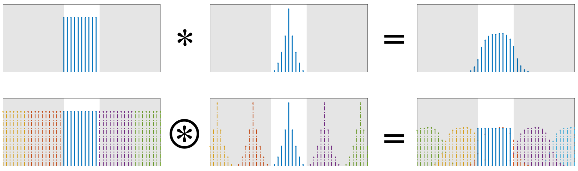

Fig. 3: Cyclic convolution vs Acyclic convolution. Note that the $*$ symbol represents the acyclic convolution operator while the $\circledast$ represents the cyclic convolution operator

Understanding the time-domain product of signals as the convolution of their spectra allows us to quantify aliasing by comparing the out-of-band energy to the in-band energy in a practical way by performing acyclic convolution of spectra instead of time-domain product of waveforms. This corresponds to the analytical concept of Aliasing Signal-To-Noise Ratio (ASNR) [6]. The only missing part of this puzzle is how to recover the control signal from a dynamic processor.

Methodology

The operation of a dynamic processor can be described by the multiplication of the input signal $x[n]$ with a control signal $g[n]$ to produce the output signal $y[n]$, as shown in *Eq. 2*.

$$x(t) \times g(t) = y(t) \leftrightarrow X(\omega) * G(\omega) = Y(\omega) \tag{2}$$

Although sometimes it is possible to divide the output signal $y[n]$ by the input signal $x[n]$ to obtain the control signal $g[n]$; one must be careful in case the input signal is close to 0. Nonetheless, if the dynamic processor to be analyzed has a side-chain input, the experimental setup shown in *Fig. 4* allows us to recover $g[n]$ in a safe manner.

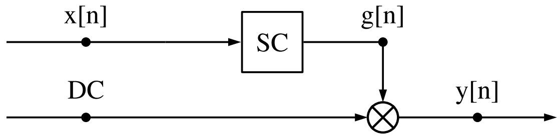

Fig. 4: Experimental Setup of Dynamic Processor with external sidechain to extract the gain control function

In this setup, the signal to be compressed $x[n]$ is fed to the external side-chain input of the dynamic processor while a constant DC signal is fed to its input, giving us *Eq. 3*. The effect of this is that at the output of the dynamic processor, we obtain the gain control signal $g[n]$ that would have been produced if the input signal had been processed. In this scenario, we have direct access to the input signal $x[n]$ and the control signal $g[n]$ before these two are multiplied in the time domain, or correspondingly, convolved in the frequency domain.

$$1 \times g(t) = g(t) \leftrightarrow \delta(\omega) * G(\omega) = G(\omega) \tag{3}$$

The only thing left is to compute their spectra, convolve them and compare the energy above the Nyquist limit to the energy below it to obtain the ASNR as described by *Eq. 4*.

$$\text{ASNR} = \frac{\int_{f_s/2}^{f_s} |Y(f)|^2}{\int_0^{f_s/2} |Y(f)|^2} \tag{4}$$

The following pseudo-code shows how to implement this measurement in a block-processing scheme for continuous measurement

```
% For each block
for i = 1:numBlocks
    % get the input signal
    x = getNextBlock(inputSignal)

    % create DC signal
    dc = ones(size(x))

    g = processWithSideChain(dc, x)

    % Compute spectra and convolve
    X = fftshift(fft(x))
    G = fftshift(fft(g))
    Y = convolve(X, G)
```

```
    % Compute Power Spectrum
    Syy = abs(Y).^2

    % sum in band and out band bins
    outBandE = sum(Syy(outBandIdx))
    inBandE = sum(Syy(inBandIdx))

    % get the ASNR for current block
    ASNR(i) = outBandE / inBandE
end
```

Note that this setup allows the measurement of ASNR as a user who may not know what is happening inside the dynamic processor. However, this approach will not work properly unless we know that there is no oversampling happening internally. In fact, there are many ways to implement oversampling, but in general, one would expect aliasing to occur both during the time-domain product of the input signal and the control signal, as well as during the decimation stage to return to the base rate, as shown in *Fig. 5*.

Nevertheless, this method is still extremely useful for a DSP engineer who has access to the internal variables during the design phase. In particular, it is useful to quantify the aliasing produced by the time-domain product independently of the aliasing produced during the decimation stage of the oversampling structure and compare it to other strategies to minimize aliasing.

Techniques to reduce aliasing

The previously described method was used to measure the aliasing produced by different compression algorithms with several features intended to control aliasing during processing.

These features include:

1. Source Band Limiting (SBL): implemented through a 12-*th* order elliptic filter with 0.01 dB pass-band peak-to-peak ripple and 120 dB minimum stop-band attenuation with a cutoff frequency of 20kHz. Note that this requires roughly 50 FLOPS per sample when implemented

2. Control Signal Band limiting (CBL): implemented through a 12-*th* order elliptic filter with 0.01 dB pass-band peak-to-peak ripple and 120 dB minimum stop-band attenuation with a cutoff frequency of 2kHz. Notice that this requires roughly 50 FLOPS per sample when implemented

3. Control Signal Smoothing (CS): implemented through a simple one-pole filter (*leaky integrator*) with a time-constant τ equal to 1ms. This might seem equivalent to using longer time constants in the compresor but depending on the topology that might not be the case. For example an amplitude follower with independent attack and release times followed by a computation of attenuation could produce sharp edges as it goes into compression independently of its time constants. Accordingly, this method is closer (but not equivalent) to a soft knee. Note that his requires 5 FLOPS per sample when implemented

4. Oversampling (OS): The algorithm is wrapped with a 2X oversampling algorithm using a 140-tap long anti-imaging and anti-alias filter designed with the Parks-McClellan algorithm to achieve 120 dB of rejection with a cutoff frequency of 20kHz with only 0.01 dB of pass-band ripple. Note that this requires roughly 560 FLOPS per sample of the original base rate. This is because for the oversampling structure we need 2 of these filters and they will be running and the oversampled rate. Moreover, any processing inside the oversampled section also becomes twice as costly in terms of computation.

Experimental Setup

3 samples were used to test this. A snare sample, a drum loop and a guitar riff. *Fig. 6* shows the snare sample that was used to exemplify the value of the process. Note that the snare, as well as the other samples were amplitude normalized. The parameters of the dynamic processor (a compressor) were chosen to produce sufficient aliasing and were maintained across all tests. Nonetheless, for reference, a threshold of $-25dB$ was used, the attack time was set to $1ms$ while the release time was set to $100ms$ and lastly the compression ratio was $4:1$. Finally, although the implementation of the side chain is outside the scope of this analysis, it is based on what Giannoulis, Massberg and Reiss call *Linear domain Branching* implementation [7]. However, an equivalent analysis can be done with any other type with varying results.

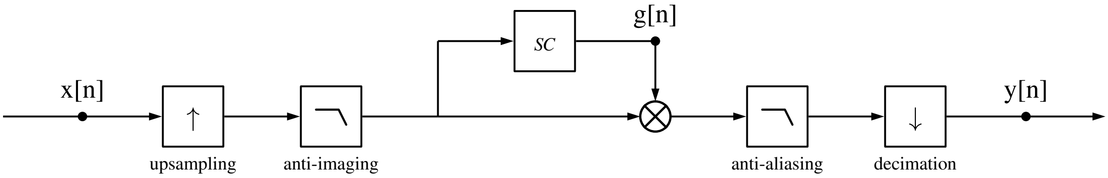

Fig. 5: Simple Feed-forward Dynamic Processor structure wrapped by an oversampling structure.

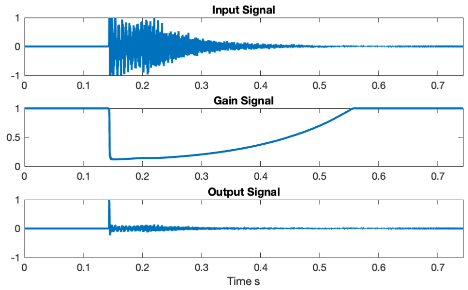

Fig. 6: Snare Hit used to exemplify the process

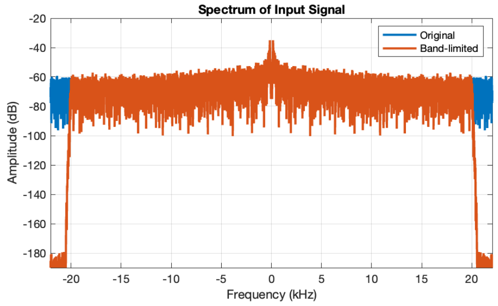

Fig. 7: Input signal before and after band-limiting filter to generate guard-band after 20kHz

Thanks to understanding the time-domain product of signals as the convolution between their spectra, band-limiting both the input signal and the control signal so that their convolved spectra does not expand beyond the the Nyquist limit is an effective way to reduce aliasing. *Fig. 7*, shows the effect of band-limiting the input signal.

Additionally, the control signal can simply be smoothed with a one-pole filter. In contrast to the band-limiting elliptic filter, the simple one-pole filter only attempts to round-off the sharp corners in the control signal produced by the side-chain algorithm (for example, with a hard knee on a compressor). This is important since these sharp corners are responsible for the slow roll-off in the spectrum of the control signal. *Fig. 8* shows the time-domain and frequency-domain

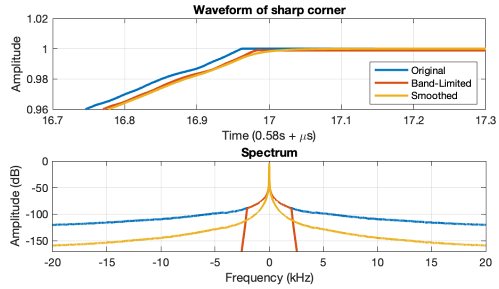

Fig. 8: Control signal after being band-limited or smoothed. The sharp corner on the waveform is smoothed out with the leaky-integrator while the elliptic filter creates an ample guard-band without changing significantly the waveform

effects of band-limiting and smoothing the control signal zooming in exactly at the moment where the compressor stops attenuating the signal right at the end of its release near 0.58 seconds.

The compression algorithm was implemented including all the possible combinations of the previously described features (band-limiting the source signal (SBL), band-limiting the control signal (CBL), smoothing the control signal (CS) and wrapping the structure with a 2X oversampling (OS)) so *Table. 1* shows a summary of the produced measurements with said arbitrarily chosen samples together with the computation cost of such implementation expressed in FLOPS per processed sample.

Results

Being able to produce a method to measure aliasing seems to be a necessity to properly design dynamic processors. This is specially apparent from the example provided, since not all the features provided the same benefit according to the measurements. Moreover, having a way to measure aliasing, we might ask if oversampling is always the right approach. *Table. 1* shows a tremendous improvement (of close to 50

SBL	CBL	CS	OS	Snare ASNR (dB)	Drums ASNR (dB)	Guitar ASNR (dB)	FLOPS
				-47.38	-80.49	-82.96	20
✓				-61.01	-93.29	-83.80	70
	✓			-49.82	-82.90	-100.27	70
✓	✓			-99.54	-127.57	-134.70	120
		✓		-50.05	-83.51	-111.04	25
✓		✓		-101.16	-129.70	-123.41	75
	✓	✓		-51.32	-84.16	-112.13	75
✓	✓	✓		-112.69	-129.80	-153.16	125
			✓	-75.63	-93.29	-83.80	600
✓			✓	-87.09	-86.44	-84.82	700
	✓		✓	-78.31	-127.57	-134.70	700
✓	✓		✓	-106.82	-135.92	-141.14	800
		✓	✓	-78.39	-129.70	-123.41	610
✓		✓	✓	-122.88	-125.21	-124.59	710
	✓	✓	✓	-79.8	-129.80	-153.16	710
✓	✓	✓	✓	-124.07	-129.83	-154.73	810

Table 1: Measured ASNR for a snare, a drum loop and a guiar riff samples with different combinations of techniques to reduce aliasing during compression

dB) in the measured ASNR by combining simple techniques that do not need at all multi-rate algorithms such as oversampling techniques. Moreover, when compared to the estimated computation cost, the oversampling structure by itself is 4.8 times more expensive when compared to the combination of the 3 proposed techniques having an ASNR between 40 and 50 dBs higher. On the other extreme, the combination of all the techniques produced at the most only 12 dB less ASNR while being 6.5 times more costly.

Discussion

In general, a DSP engineer might assume that using more and more oversampling will provide significantly better aliasing rejection. However, how can we know if that is the case if we do not have a way to measure aliasing?

It is crucial to consider that the amount of produced aliasing is directly related to the source material being processed. It is not the same to compress a sinusoid than a noise burst. Broad experimentation with these algorithms is needed during the design phase to properly evaluate the advantages and disadvantages of different processing schemes. Moreover, because of the dependency on the signal, these results should not be understood as a general behavior but rather as an example of why it is important to assess ASNR and consider how different implementations might affect it.

On the other hand, many times the final user of a processor is provided with controls to enable or disable oversampling without truly knowing the extent of these changes. Embedding this measurement as part of a processor's visual feedback could provide information about the algorithm's performance. This would better inform the final user about whether it is worth engaging certain features, like oversampling, in a dynamic processor depending on the source material to be used.

Additionally, although the measurements provided still show a simple oversampling wrapper as a relatively effective way to reduce aliasing, yet it is surprising to notice that the other three proposed techniques achieve better rejection than just oversampling at a much smaller computational cost. This does not mean that these results will hold for every material or that it is not possible to optimize each of them in different ways to achieve better performance. However, if the ASNR is calculated and tested through the design phases, better decisions can be made based on more objective data to quantify this problem. One might decide not to oversample due to the latency implications, or perhaps the computational efficiency of simply band-limiting the input signal and smoothing the control signal is enough instead of going into expensive multi-rate schemes. All of these are good reasons to make decisions instead of simply assuming that aliasing can only be solved by tremendous amounts of oversampling.

Finally, it is worth noting that this number may not necessarily match the perception of aliasing. Further work may need to be done to weight the ASNR to have a more perceptually meaningful description of the problem. Nonetheless, it is worth mentioning that under informal listening tests, the combination of the three initial strategies was effective in reducing a "metallic" quality on the compressed signal when no strategy was used or even compared to only using oversampling.

Conclusions

Although aliasing in dynamic processors is a well-known phenomenon, the lack of a measurement procedure to quantify it is a significant barrier in the design process. As DSP designers, we understand the mechanisms and fundamentals behind this process; however, the inability to quantify it implies that we are blind to how certain decisions can improve or worsen this behavior. Therefore, a metric like ASNR, which specializes in understanding aliasing, is crucial as it differs from common tools like Total Harmonic Distortion plus Noise (THD+N) or other traditional metrics, by tackling the specific issue of aliasing in digital systems.

In this example, we showed that although an oversampling structure was effective in reducing ASNR, the other three proposed strategies (band-limiting the source signal, band-limiting the control signal, and smoothing the control signal) were as effective while still being significantly less computationally costly. It may not be the case with every signal, but this finding raises questions about whether oversampling is always the ideal solution when dealing with aliasing in dynamic processors.

Furthermore, we need to recognize that measuring the produced aliasing when processing a signal with a dynamic processor is not quantifying the processor's performance by itself. It is instead quantifying the processor's performance when excited with a specific signal. Therefore, it is important not to consider these measurements as inherent properties of the processor. After all, aliasing is a phenomenon that directly depends on the source signal. Thus, these measurements should primarily be used as comparison points connected to the specific signal to be used. Nonetheless, having a way to quantify this phenomenon is an essential improvement to analyze and understand digital dynamic processors.

References

[1] Oppenheim, A. V., Schafer, R. W., Yoder, M. A., and Padgett, W. T., *Discrete-time signal processing*, Pearson, Upper Saddle River, NJ, 3 edition, 2009.

[2] Kester, W., editor, *Mixed-signal and DSP design techniques / by the technical staff of Analog Devices*, Analog devices series, Newnes, Amsterdam, 2003, ISBN 0750676116.

[3] Kahles, J., Esqueda, F., and Välimäki, V., "Oversampling for Nonlinear Waveshaping: Choosing the Right Filters," *Journal of the audio engineering society*, 67(6), pp. 440–449, 2019, doi:https://doi.org/10.17743/jaes.2019.0012.

[4] Mapes-Riordan, D., "A worst-case analysis for analog-quality (alias-free) digital dynamics processing," *Journal of the Audio Engineering Society*, 47(11), pp. 948–952, 1999.

[5] Foti, F., "Aliasing Distortion in Digital Dynamics Processing, the Cause, Effect, and Method for Measuring It: The Story of" Digital Grunge!"," in *Audio Engineering Society Convention 106*, Audio Engineering Society, 1999.

[6] Thornburg, H., *Antialiasing for nonlinearities: Acoustic modeling and synthesis applications*, Ann Arbor, MI: Michigan Publishing, University of Michigan Library, 1999.

[7] Giannoulis, D., Massberg, M., and Reiss, J. d., "Digital Dynamic Range Compressor Design—a Tutorial and Analysis," *Journal of the audio engineering society*, 60(6), pp. 399–408, 2012.

[8] Smith, J. O., *Mathematics of the Discrete Fourier Transform (DFT)*, http://ccrma.stanford.edu/~jos/mdft/, accessed (Jan 20 - 2023), online book, 2007 edition.

[9] Smith, J. O., *Spectral Audio Signal Processing*, http://ccrma.stanford.edu/~jos/sasp/, accessed (Jan 20 - 2023), online book, 2011 edition.