Chapter 6. Distortion

Distortion in music appears when the peaks of the audio signal are compressed. In the analog world, this can happen when an amplifier gain is increased above certain levels and the signal begins to overload internal circuits. Digital distortion is similar.

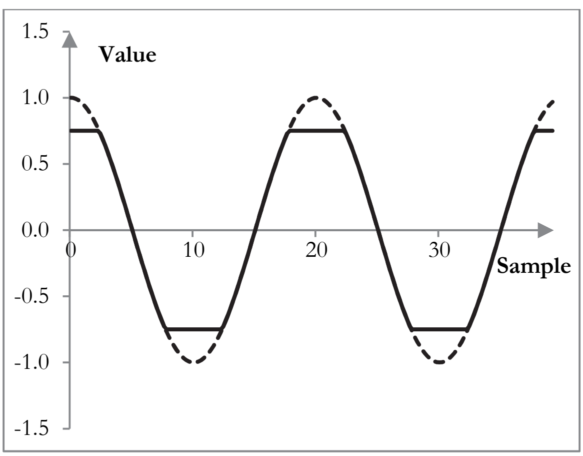
6.1. Distortion

Consider the following computation of the output signal $y(k)$ from the input signal $x(k)$.

$$y(k) = \begin{cases} 0.75, & |x(k)| \geq 0.75 \\ x(k), & |x(k)| < 0.75 \end{cases} \tag{6.1}$$

This operation is known as a *hard clip*. It cuts off the peaks of a simple sine wave with peak amplitude of 1 as in figure 18. The operation is simple, but this is not the end of the story. The resulting wave is no longer a simple wave and we should expect that its frequency content is different than that of a simple wave.

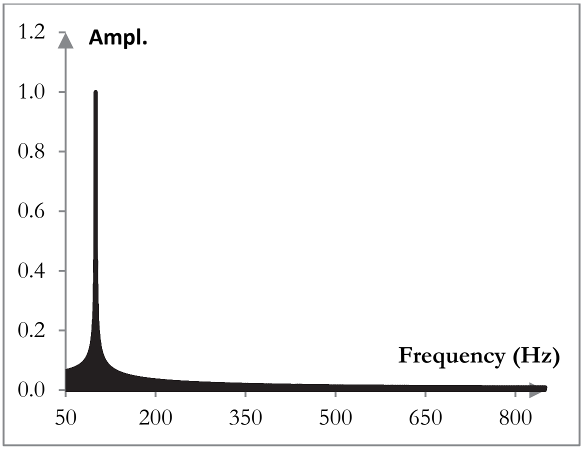
Figure 18. Hard clip of a simple wave



This simple wave with frequency of 100 Hz and peak amplitude of 1 is sampled at 2000 Hz. The wave is then truncated, whenever its values exceed 0.75.

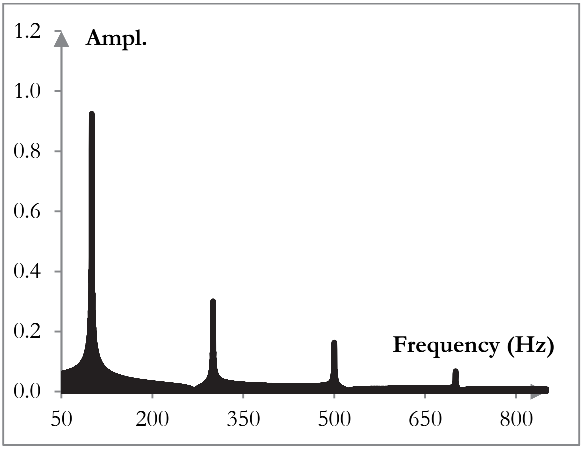
Figures 19 and 20 show the frequency contents of the original simple wave and of the modified signal. While the input simple wave has only one peak at 100 Hz – the frequency of the simple wave – the distorted signal has additional notches at 300 Hz, 500 Hz, and 700 Hz. These frequencies are odd integer multiples of 100 Hz and are called *odd order harmonics* of 100 Hz.

Figure 19. Frequency content of the original signal



The frequency content of the original signal has only one peak, as there is only one simple wave in that signal.

Figure 20. Frequency content after the hard clip



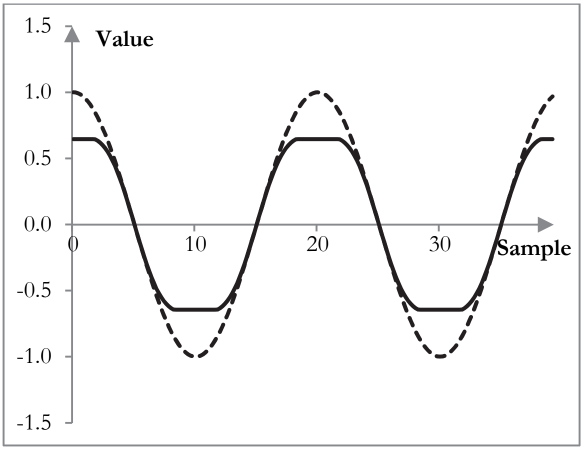
Hard clipping lowers the amplitude of the original wave and introduces odd order harmonics.

Distortion introduces harmonics. In fact, any nonlinear modification of the signal introduces harmonics. A nonlinear modification to the signal can then be called *harmonic distortion*. The more severe the clipping is (e.g., 0.5 instead of 0.7), the more accentuated the harmonics are.

A sharp truncation of the signal peaks is not the only way to produce distortion. A softer distortion can be introduced by slowly compressing and flattening of the peaks. The *cubic soft clipper* is a well-known function that does exactly that.

$$y(k) = \begin{cases} -\left(w - \dfrac{w^3}{3}\right), & x(k) < w \\ x(k) - \dfrac{x(k)^3}{3}, & |x(k)| \le w, \quad w > 0 \\ \left(w - \dfrac{w^3}{3}\right), & x(k) > w \end{cases} \tag{6.2}$$

Figure 21. Cubic soft clipper



The cubic soft clipper rounds the clipped peaks.

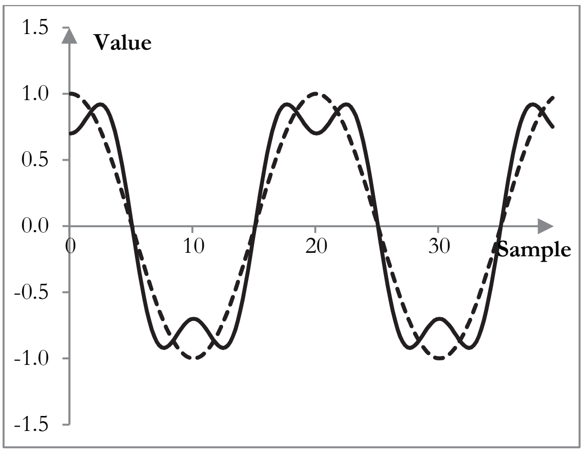
With larger w, the clipping becomes softer and closer to the peak of the wave. The cubic soft clipper introduces similar odd order harmonics in the signal.

There are many operations that we can use to clip the signal similarly. Clipping, however, need not always occur. The following distortion operation compresses the signal peaks without actually clipping them. It introduces the same odd order harmonics, although with lower energy.

$$y(k) = atan(x(k)) \tag{6.3}$$

It is easy to explain why clipping introduces odd order harmonics. Consider figure 22, which shows a signal at 100 Hz with peak amplitude of 1 (dashed) and the sum of the signal at 100 Hz with a signal at 300 Hz with amplitude of 0.3 (solid). The impact of adding a simple wave with three times the frequency and lower amplitude is very similar to the impact of clipping the wave.

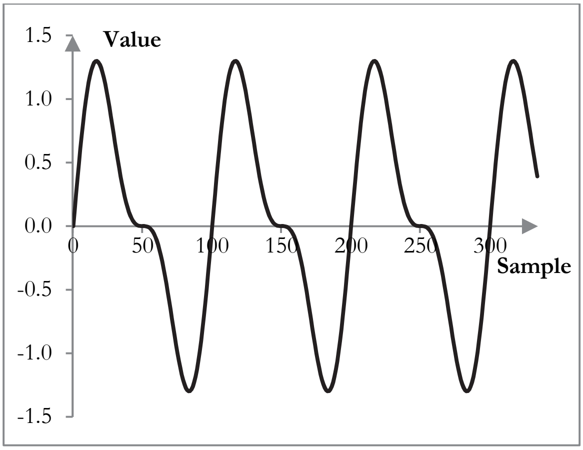
Figure 22. The impact of odd order harmonics



Adding the first odd order harmonic at three times the frequency compresses the peaks of the wave similarly to clipping.

If we wanted to introduce *even order harmonics* – even integer multiples of the original frequency – we could examine the impact of adding two waves, one of which is, say, twice the frequency of the other one – the first even order harmonic. A signal composed of two such waves is shown on figure 23. Note that even order harmonics do not compress, but expand the signal peaks and troughs. Further, the signal becomes asymmetric – unlike adding the first odd order harmonic, which produces a rather symmetric signal.

Figure 23. The impact of even order harmonics



Adding the first even order harmonic at two times the frequency does not compress the wave peaks, but expands them.

One modification that is a good candidate for introducing even order harmonics is

$$y(k) = x(k)^3 \tag{6.4}$$

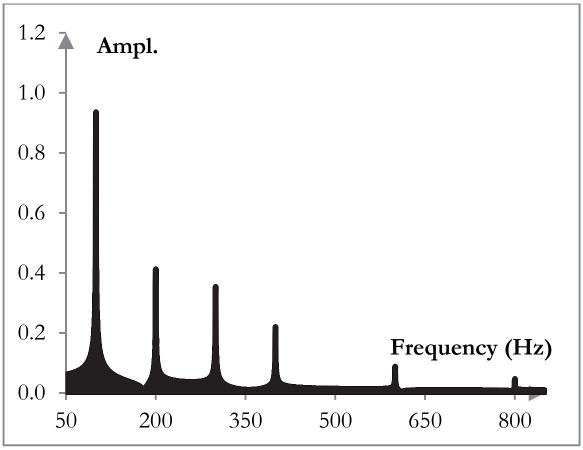
You can verify that this function introduces both even and odd order harmonics. Clipping this signal would produce a very different distortion.

The following modification could also be of interest.

$$y(k) = \begin{cases} x(k), & x(k) > x(k-1) \\ x(k)^3, & x(k) \le x(k-1) \end{cases} \tag{6.5}$$

This operation introduces only even order harmonics, except for the first odd order harmonic at three times the original frequency, as shown in figure 24. The issue with this operation is that it has to look into the past at previous samples of the signal. With the exception of this very last operation above, and unlike most operations that we will discuss in the following chapters, distortion does not look back at previous input samples. It is thus called a *memoryless* operation.

Figure 24. Frequency content of an expansion



Expanding the wave peaks introduces even order harmonics. The fact that there is a peak also at 300 Hz, means that the first odd order harmonic is still present. The remaining peaks are at 200 Hz, 400 Hz, etc.

Digital distortion in practice could be as simple as the operations above or more complex. For example, distortion could require upsampling[1] of the audio data. It may be necessary, because harmonics of high recorded frequencies may be higher than the Nyquist-Shannon frequency for the original sampling rate. *Upsampling* recreates the signal, but at a higher sampling rate, at which the Nyquist-Shannon frequency would also be higher. Digital distortion may also include an equalizer, so that the magnitude of harmonics is controlled. Finally, distortion may distort different parts of the frequency spectrum differently, by splitting the original signal into several parts, say, low, middle, and high frequencies.

6.2. Upsampling and downsampling

An easy example of upsampling is to resample the signal at twice the original sampling frequency, by estimating the value of the signal in the middle of any two samples. If the value of the signal at sample k is $x(k)$ and the value of the signal at sample $k + 1$ is $x(k + 1)$, then the value of the middle, using simple linear interpolation, is

[1] Sometimes upsampling is up to eight times the original sample rate.

$$x(k+0.5) = \frac{x(k) + x(k+1)}{2} \tag{6.6}$$

In general, if we wanted to upsample by an integer multiple N of the original frequency and we use linear interpolation, then the samples between k and $k + 1$ would be computed as follows

$$x\left(k + \frac{n}{N}\right) = x(k) + \frac{n}{N}\big(x(k+1) - x(k)\big) \tag{6.7}$$

$n = 0, 1, \ldots, N - 1$. An alternative is to use polynomial interpolation. If we know, for example, that the signal at points $k - 1$, k, and $k + 1$ is $x(k - 1)$, $x(k)$, and $x(k + 1)$, then we can compose a polynomial $P(k)$ of degree at least 3

$$P(k) = a_3 x^3 + a_2 x^2 + a_1 x^1 + a_0 \tag{6.8}$$

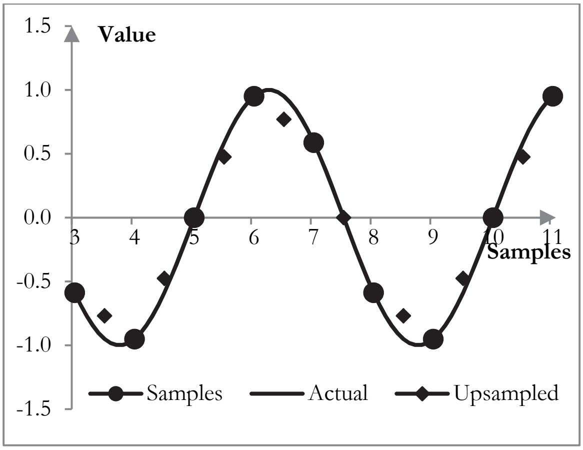
that passes through all three points by solving the system of equations

$$\begin{aligned} a_3(k-1)^3 + a_2(k-1)^2 + a_1(k-1) + a_0 &= x(k-1) \\ a_3 k^3 + a_2 k^2 + a_1 k + a_0 &= x(k) \\ a_3(k+1)^3 + a_2(k+1)^2 + a_1(k+1) + a_0 &= x(k+1) \end{aligned} \tag{6.9}$$

We can then compute the value at any new sample between the original samples from the polynomial, although this may be too much work.

Other types of interpolation exist. Whichever one we use, we will be imprecise. It is impossible to know the exact value of a sampled signal at points at which it was not originally sampled. Since, during upsampling, we have increased the sampling frequency from f_s to some other sampling frequency $f_{s,1}$, we have also increased the Nyquist-Shannon frequency from f_s / 2 to $f_{s,1}$ / 2. Since upsampling is imprecise, we must be careful not to introduce frequencies in the upsampled signal that do not exist in the original signal, namely the frequencies from f_s / 2 and above. We must then use low pass filters – operations that filter out the frequencies above f_s / 2, while preserving the frequencies below f_s / 2.

Figure 25. An example of upsampling



Since we do not know more than just the values at the samples at the current sampling rate, upsampling cannot be precise. This upsampling of a simple wave at 300 Hz from the sampling frequency 2000 Hz to 4000 Hz will introduce harmonics. It will, specifically in this example, introduce the frequency of approximately 1700 Hz into the signal.

Occasionally, upsampling may be at a sampling frequency that is not an integer multiple of the current sampling frequency, say 1.5 times the current sampling frequency. In this case one might first upsample to three times the current sampling frequency, and then downsample to half of the new sampling frequency.

Downsampling works very similarly. If we downsample from the sampling frequency f_s to $f_{s,2}$, then we need to be careful to remove the frequencies between $f_{s,2}$ and f_s. This time, we must filter those frequencies out before the downsampling process. If these frequencies are not filtered out, they will be confused for lower frequencies below the new sampling frequency, which, as explained in Chapter 3, is known as aliasing. Thus, the low pass filter that will be used to remove such frequencies is sometimes called an *anti-aliasing* filter. Since it is best to downsample by integer divisors of the current frequency, sometimes we might upsample first, before we downsample.