Chapter 4

Aliasing, Sampling & Reconstruction

4.1 Aliasing

Alias:

1. An assumed name
2. (computers) An alternate name or address, especially an e-mail address that forwards incoming e-mail to another address.
3. (signal processing) An spurious signal generated as a technological artifact.

From Nyquist's sampling theorem, we know that when we sample an analog signal $f(t)$ at a sampling rate of f_s samples per second ($= \frac{\omega_s}{2\pi}$), we should ensure that all components of the signal lie within the range $\pm\frac{1}{2}f_s\,\mathrm{Hz}$.

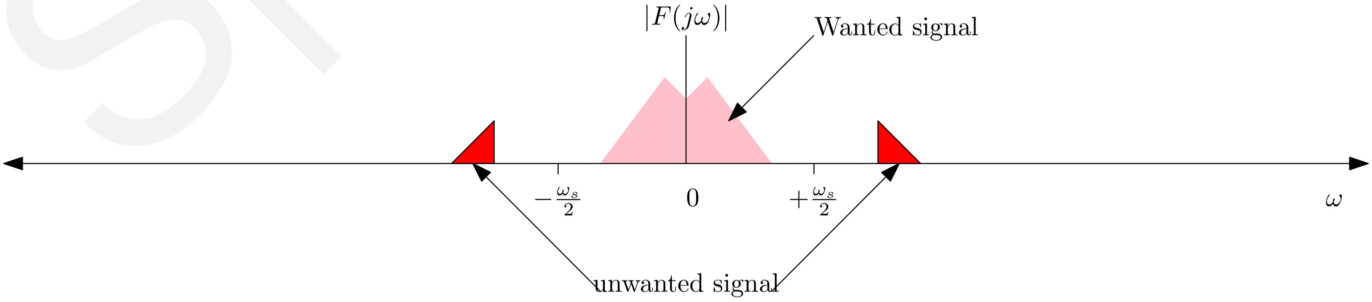

Figure 4.1.1: Analog signal before samping with some unwanted components outside of the allowable frequency range (for sampling at a rate of f_s Hertz).

If there are components that lie outside of that range, e.g. see Figure 4.1.1, they will appear within that range after sampling - a process known as *aliasing* or *folding*. This process is illustrated in Figure 4.1.2.

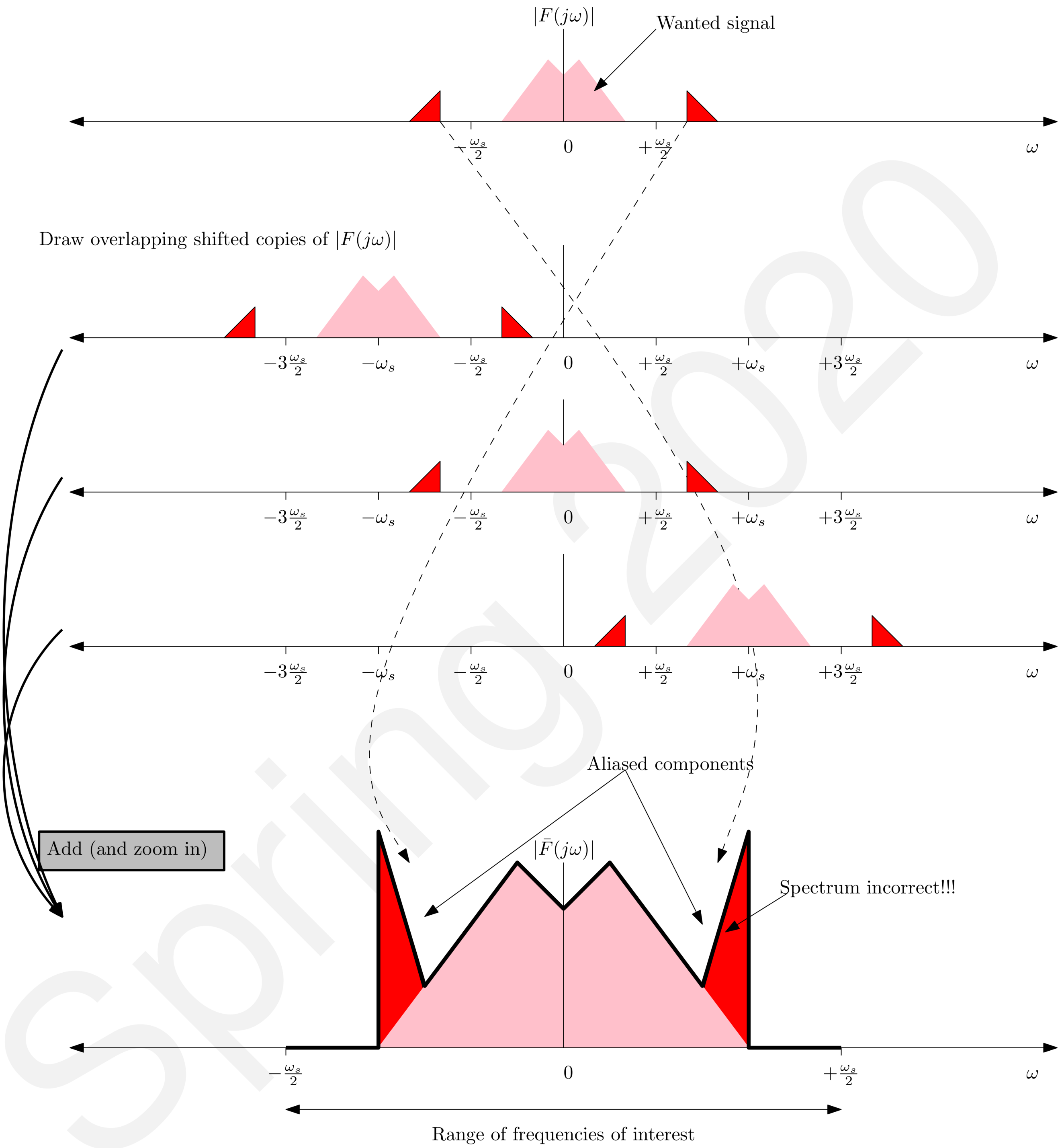

Figure 4.1.2: Spectrum after sampling. The red parts have now appear to have a lower frequency than before - i.e. they are *aliased*.

Exercise:

Try to get the same result using the frequency domains plot as before.

4.1.1 Folding - another name for aliasing

We can think of the process of adding shifted version of the frequency spectrum together as being like folding the spectrum on top of itself as illustrated below:

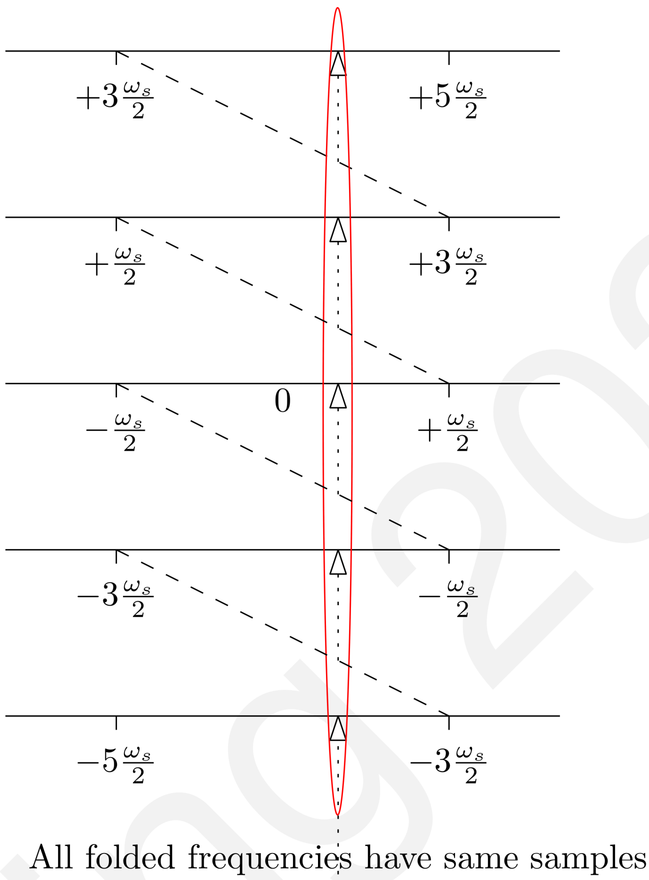

Figure 4.1.3: Spectrum folding after sampling. If we fold the analog spectrum in this way all signal components the "line-up" vertically will appear to have the same frequency after sampling.

4.1.2 Sine wave example

Consider an analog sine wave signal having frequency f_o, i.e. $f(t) = \sin(2\pi f_o t)$. Imagine it is sampled at rate f_s samples per second resulting in the stream of sample values, $f_n \triangleq f(nT)$, where $T = \frac{1}{f_s}$ is the time interval between samples.

Show that:

$$f_n = \sin(2\pi(f_o + mf_s)nT) \quad \forall m \in \mathbb{Z}$$

Solution:

$$
\begin{aligned}
f_n &\triangleq f(nT) \quad \forall n \in \mathbb{Z} \\
&= \sin(2\pi f_o nT) \\
&= \sin(2\pi f_o nT + 2\pi mn) \quad \forall m \in \mathbb{Z} \\
&= \sin\left(2\pi f_o nT + 2\pi mn\frac{1}{T}T\right) \\
&= \sin(2\pi f_o nT + 2\pi m f_s nT) \\
&= \sin(2\pi (f_o + m f_s) nT)
\end{aligned}
$$

The importance of this is that if the frequency of the sine wave is f_o,or $f_o \pm f_s$, or $f_o \pm 2f_s$, or $f_o \pm 3f_s$ etc..., the values of the samples f_n are going to be the same!
This is a time domain example of how all the frequencies $f_o + mf_s$ for $\forall m \in \mathbb{Z}$ alias back to f_o. This is illustrated in Figure 4.1.4.

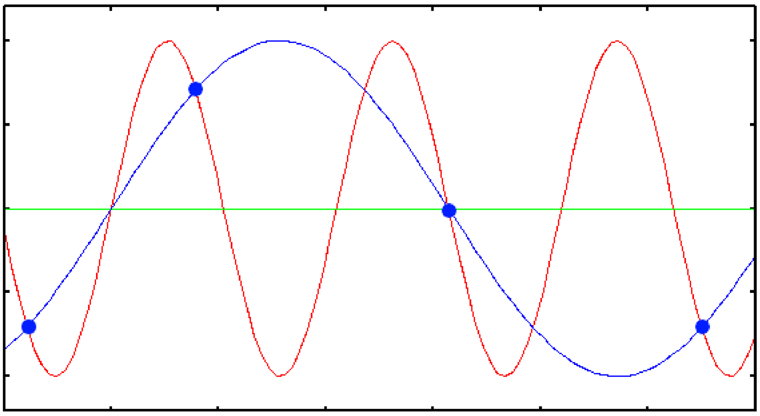

Figure 4.1.4: Two analog sine waves having frequency f_o(blue) and $f_o + 3f_s$ (red) both have the same sampled values (the dots) when sampled at a rate of f_m samples/sec.

4.2 Sampling in practice

4.2.1 Anti-Aliasing filter

As per Nyquist's sampling theorem we know that the signal presented to an Analog-to-Digital (ADC) must not contain frequency components beyond $\pm\frac{1}{2}f_s$. Sometimes (but rarely) this is the case naturally, but usually we need to do some analog signal processing, e.g. filtering, to ensure that this is (at least approximately) true. Specifically there is almost always an analog anti-aliasing filter positioned prior to the ADC.

Another reasons is that in many systems noise is present and it can have a much larger bandwidth than the wanted signals - this noise will also fold during the sampling process and if not removed it can dominate and the signal can be destroyed forever => we almost always have an anti-aliasing filter.

The ideal anti-aliasing filter has the following characteristics:

1. Passes all frequencies between $\pm\frac{1}{2}f_s$ Hertz (gain 0dB)
2. Does not distort the signals passed - just a delay
3. All other frequencies are prevented from passing.

The type of filter is known as a "*brick-wall*" specification and is theoretically impossible!!!

Real-world low-pass filters have the following characteristics:

1. This is some pass-band distortion
2. The stop-band does not attenuate the signals to zero completely
3. There is a transition band between the pass and stop bands

See 4.2.1 for a typical low pass filter specification.

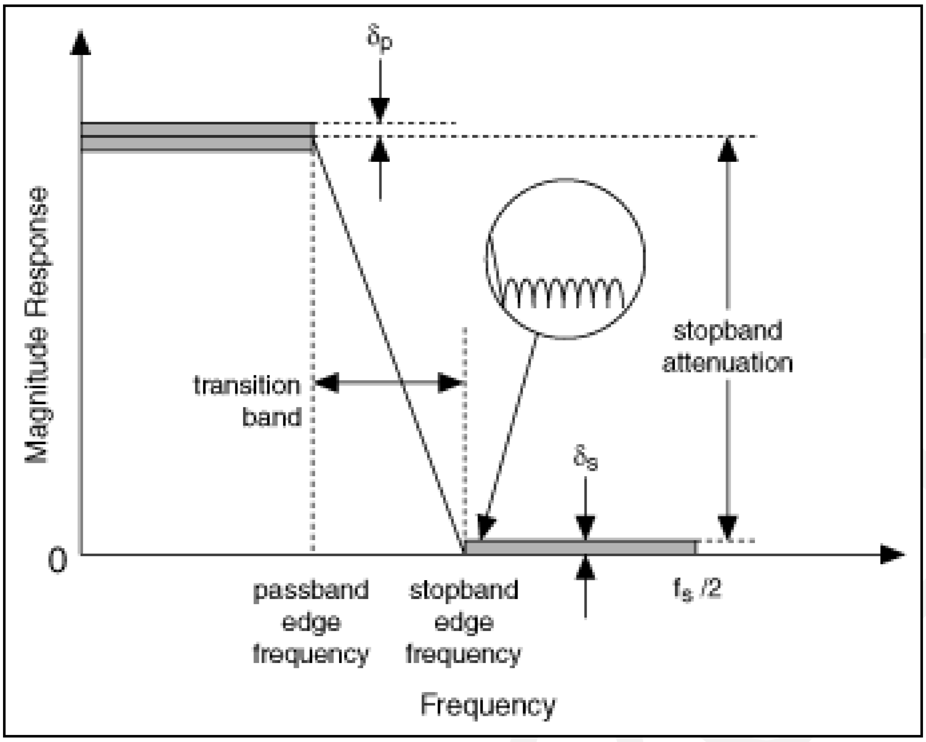

Figure 4.2.1: A real world low-pass filter specification.

Q. So what can we do?

A. Oversample.

4.2.1.1 Oversampling

In theory a signal having its highest wanted frequency component at f_{max} (plus some other unwanted components) can be filtered by a brick-wall filter having cut-off frequency f_{max} and then sampled at a rate of $2 \times f_{max}$.

In practice, as brick-wall filters don't exist, we use a real world low pass filter with a pass-band edge frequency > f_{max}, and we sample at a rate $f_s > 2 \times f_{\text{stopband}}$, where f_{stopband} is the stop-band edge frequency ($> f_{max}$). This way the signal will be faithfully captured as too will some of the unwanted signal components - but importantly they will not interfere with the wanted parts.

Note that the sampling here is now $> 2 \times f_{max}$, hence the name *oversampling*.

See Figure 4.2.2 for an example. It is important to note that the sampling process in this case converts both wanted and unwanted signals but this is done in such a way that they don't interfere with each other. Perhaps the subsequent digital processing will further "clear-up" the signal.

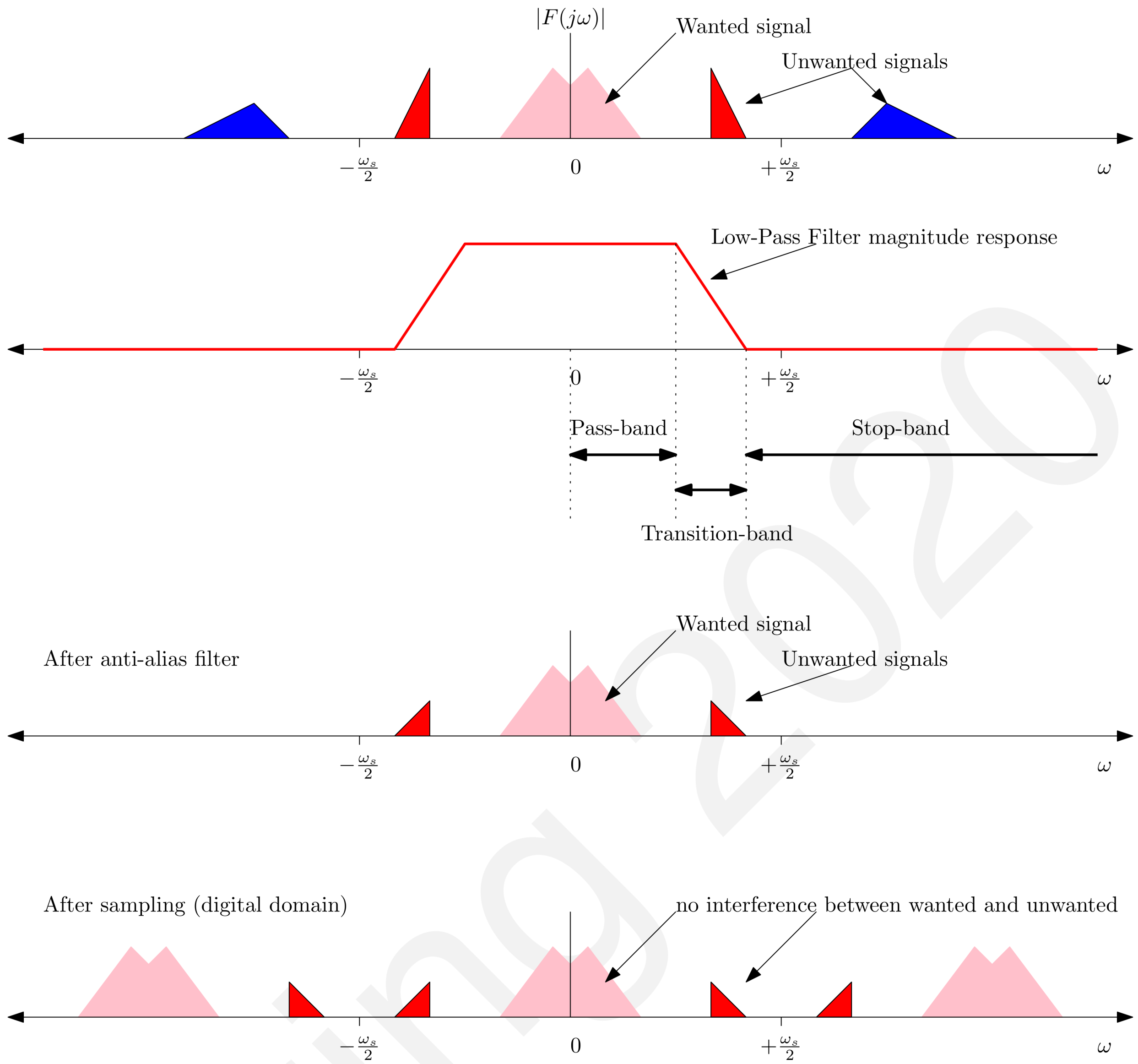

Figure 4.2.2: Real world sampling using a non-ideal low-pass anti-aliasing filter.

Note:

Analog filters can be big and bulky and can consume a lot of power, and the tighter the specification the bigger they become => we usually want to design small easy-to-implement anti-aliasing filters. This can be done by increasing the sampling rate higher, i.e. by over-sampling by a large factor. Today it is not uncommon to sample at $10 \times f_{max}$ instead of the $2 \times f_{max}$ that Nyquist would suggest.

4.2.2 Sample & Hold

Impulse sampling is NOT possible as it requires the generation of Dirac delta functions which are not possible[1].

[1] Not possible due to their infinite amplitude, and infinitesimally small duration.

We can however implement circuits (using transistors and capacitors) that can approximate well a “sample & hold” (S/H) function resulting in a waveform like the one shown in Figure 4.2.3.

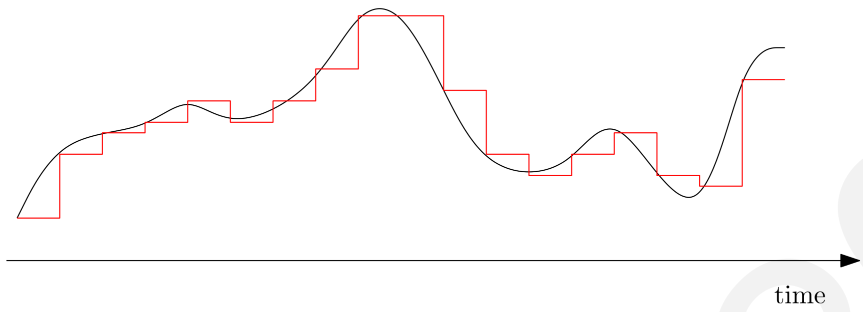

Figure 4.2.3: Analog signal input (smooth line) to a S/H device and the corresponding output (red discontinuous line).

If we follow this S/H operation with a device that converts the analog voltage in each time step to a numeric value (the core of the ADC), then we’ll have a series of numbers which represent the value of the signal at the sample instances. For this to work correctly the following need to be true:

1. The value held on the S/H output must represent the voltage at the exact sampling time instant
2. The value held must be stable for the duration of the conversion

It is clear that the sampling rate in this scheme is limited by the speed of the conversion.

4.2.2.1 ADC chain

Based on the above discussions we can now draw the entire ADC chain as per Figure 4.2.4.

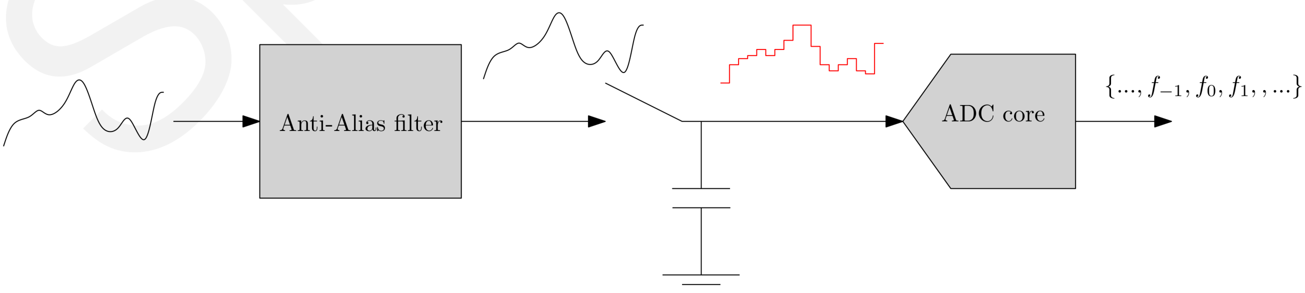

Figure 4.2.4: Typical Analog to Digital Conversion (ADC) chain.

4.3 Signal Reconstruction

Given a set of signal samples $\{f_n\}$ how can we reconstruct the corresponding analog signal? Well we saw in Section 3.2.2.2 that *perfect reconstruction* can be achieved with an ideal low-pass filter, known as a reconstruction filter. However, just as in the ADC case, if the sampling rate $f_s = 2 \times f_{max}$, then this filter needs to be a non-realizable brick-wall filter - not to mention that it is not possible to generate input to said filter (a train of delta impulse functions).

The solution is to do exactly as per the ADC above, but in reverse.

1. Have an DAC circuit that can generate a S/H type analog signal out based on a discrete series of numeric value (the $\{f_n\}$)
2. These are held constant for the sampling interval
3. This is applied to a non-ideal reconstruction filter.

This is shown in Figure 4.3.1, where it is important to note that the sampling rate $f_s > 2\times f_{max}$.

Also shown in Figure 4.3.1 is a mathematical way of modelling the entire system. In particular it is important to note that the final spectrum is that of the sampled spectrum (i.e. with all the repeated spectra) multiplied by a sinc response (arising from the rectangular impulse repsonse of the DAC core) and the reconstruction filter.

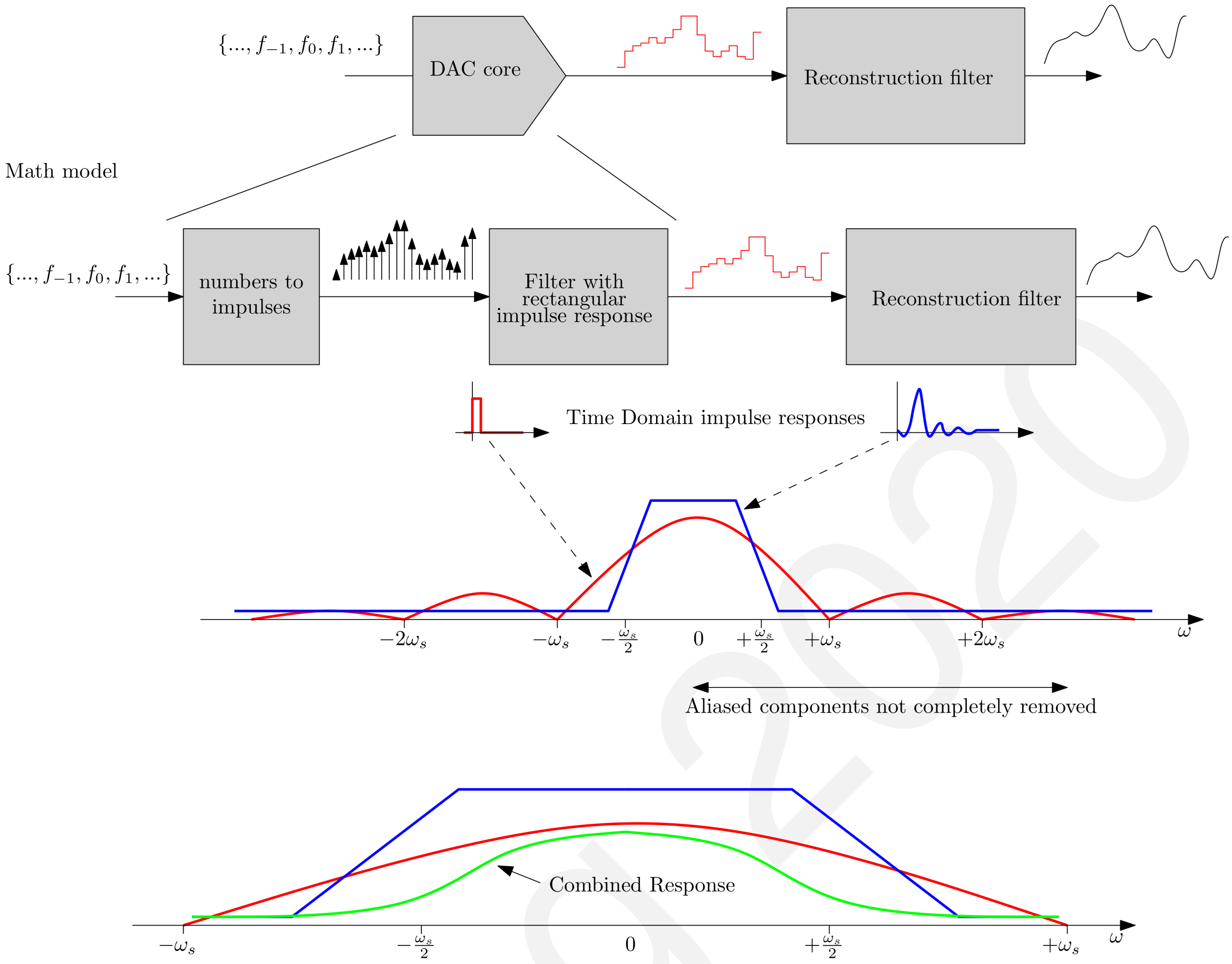

Figure 4.3.1: Typical Digital to Analog Conversion (DAC) chain.

4.3.1 Reconstruction filter

The reconstruction filter should have the following properties

1. The response, when combined with the S/H sinc response, should be ideal at the wanted signal frequencies
2. This combined response should be (ideally) zero at the aliased frequencies.

If the oversampling rate is high, then the above conditions are relatively easy to approximately achieve.

However if operating close to the theoretical $2 \times f_{max}$ then this becomes very difficult - sometimes the digital signal itself is pre-distorted so that the combined response of the pre-distortion, sinc filter, and reconstruction yields the desired behavior. This is an advanced topic.

4.3.2 Quantizer

After sampling, the signal is discrete in time, but is still a continuum in amplitude.

- The signal needs to be mapped (i.e. rounded) to discrete levels
- This process is called quantization, see figure 4.3.2

As can be seen from Figure 4.3.2, quantization introduces irreversible rounding errors, i.e. once done can never be undone, and these can impact system performance, e.g. cause an audible noise in a speech signal. The following section will quantify this noise.

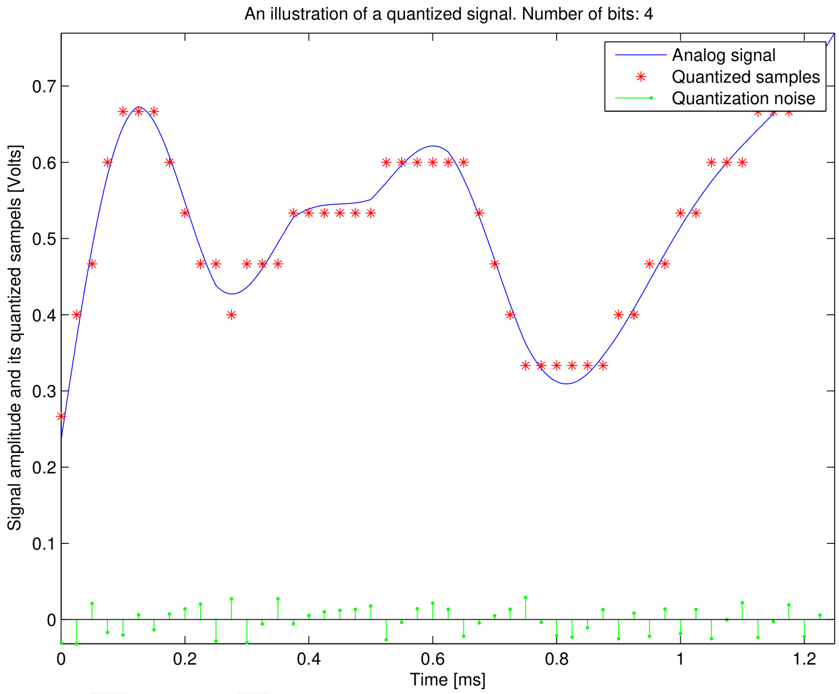

Figure 4.3.2: An analog signal sampled and quantized

4.3.2.1 Quantization noise

Taking the units of amplitude as being Volts, and by assuming a *uniform spacing*, we can define Δ Volts as that voltage separating adjacent quantized levels, see figure 4.3.3. Note that if binary words are associated with quantization levels, then a input change of Δ Volts causes a Least Significant Bit (LSB) change in the resulting binary word, hence Δ is known as the LSB Voltage.

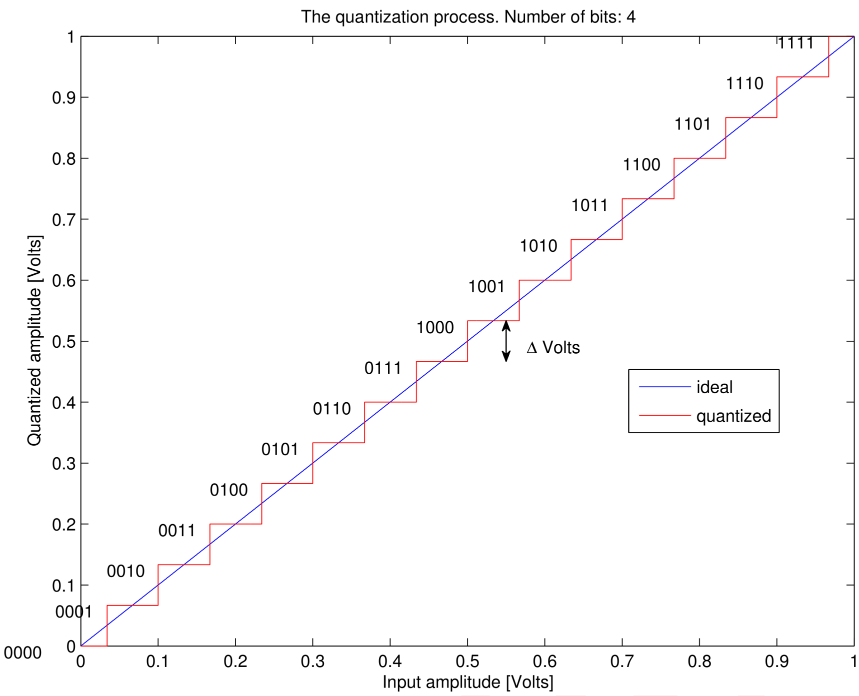

Figure 4.3.3: Typical input-output response of an ADC

We can consider the quantization noise as being equivalent to adding a random variable (RV) to the sampled signal. This random variable has the following characteristics:

- Its magnitude is never greater than $\pm{}^{\Delta}/_{2}$
- Any value in the range $\pm{}^{\Delta}/_{2}$ is possible
- Each of its possible values are equally likely

Basically it is a uniformly distributed RV on the range $\pm{}^{\Delta}/_{2}$, and so its probability density function (pdf) is:

$$f\left(n\right) \quad = \quad \begin{cases} \frac{1}{\Delta} & \text{for } -{}^{\Delta}/_{2} < n < {}^{\Delta}/_{2} \\ 0 & \text{elsewhere} \end{cases}$$

We can now calculate the Root Mean Squared (RMS) voltage of the noise introduced by the quantization process by noting that:

$$\text{RMS Noise Voltage} \quad = \quad \sqrt{E\left[n^2\right]}$$

where $E\left[\cdot\right]$ is the expectation operator (i.e. mean). It is a well known result in probability and statistics (see Theorem 2 in Appendix A) that:

$$E\left[n^2\right] = \frac{\Delta^2}{12}$$

Thus:

$$\text{Noise Power} = \frac{\Delta^2}{12} \text{ Watts}$$

We note that the noise power depends only on Δ, i.e. the separation between the quantization level, and does not depend on the signal level.

4.3.2.2 Linear model

In theory, of course, the quantization noise is correlated to the signal - but if the signal amplitude changes randomly by amounts that are large compared to Δ then it can be the case that the quantization noise can be considered as additive white noise that is uncorrelated from the signal.
This is especially true if there is Gaussian thermal noise, with a variance $\gg \frac{\Delta^2}{12}$, present in the original signal.
Note that the quantization noise is however uniformly distributed (not Gaussian).
A linear noise model, as illustrated in Figure 4.3.4 can be used.

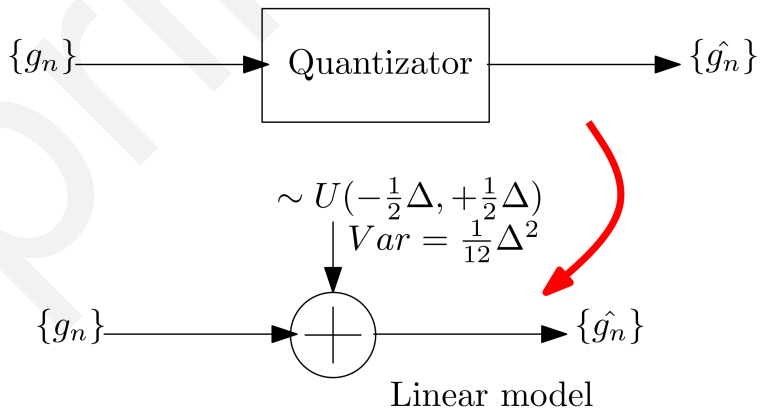

Figure 4.3.4: Approximate linear noise model for quantization noise.

4.3.2.3 Signal to Noise Ratio due to Quantization

A figure of merit often associated with an ADC is its Signal to Noise Ratio (SNR) normally measured when the largest allowable sine-wave is applied. Assuming the only source of noise is Quantization, then we can derive an analytical expression for the SNR.

It the ADC can take any input between $\pm V_{peak}$, then the largest possible sine wave has amplitude V_{peak} , and thus the signal power is $\frac{1}{2}V_{peak}^2$.

We can relate V_{peak} to Δ by noting that a b-bit ADC represents the full input range $2V_{peak}$ with 2^b levels, each separated by $\Delta = \frac{2V_{peak}}{2^b-1}$ Volts.

Thus the noise power is:

$$
\begin{aligned}
\frac{1}{12}\Delta^2 &= \frac{1}{12}\frac{4V_{peak}^2}{(2^b-1)^2} \\
&\approx \frac{1}{3}\frac{V_{peak}^2}{2^{2b}}
\end{aligned}
$$

Where the approximation is true when $2^b \gg 1$, as is usually the case.

Now we can easily compute the signal to noise ratio:

$$
\mathrm{SNR} = \frac{\frac{1}{2}V_{peak}^2}{\frac{1}{3}\frac{V_{peak}^2}{2^{2b}}} = \frac{3}{2}2^{2b}
$$

In communication systems it is usual to quote noise ratios in terms of dB :

$$
\mathrm{dB} = 10 \times \log_{10}(\mathrm{linear\,power\,ratio})
$$

So our SNR due to quantization becomes:

$$
\begin{aligned}
\mathrm{SNR(dB)} &= 10 \times \log_{10}\left(\frac{3}{2}2^{2b}\right) \\
&= 10 \times \log_{10}\left(\frac{3}{2}\right) + 20 \times \log_{10}\left(2^b\right) \\
&= 10 \times \log_{10}\left(\frac{3}{2}\right) + 20 \times \frac{\log_2\left(2^b\right)}{\log_2(10)} \\
&= 1.76 + 6.02 \times b \quad \mathrm{dB} \qquad (4.3.1)
\end{aligned}
$$

Observations:

- SNR improves by $6\mathrm{dB}$ with every extra bit (b)
- In practice the SNR varies according to frequency as well as other factors
- Quantization noise effects low and high signal voltages equally.
  - $\Rightarrow$Instantaneous SNR can be very bad for low signal levels.
  - This can be a problem for some signals, e.g. Audio.

4.3.2.4 Selection of quantization level

Lets begin by evaluating equation 4.3.1 for several difference numbers of bits:

#bits = b	#level $=2^b$	SNR [dB]
8	256	49.92
10	1,024	61.96
12	4,096	74
14	16,384	86.04
16	65,536	98.08
18	262,144	110.12

A word on terminology:

For the 12 bit case, we might say: "The quantization noise is 74dB down on the signal", or "The quantization is -74dB", or "The SNR due to quantization is 74dB".

The choice of how many bit to use really depends on the application. If for example, we are dealing with an audio signal that has been picked up by a cheap microphone and a noisy amplification circuit (as might be the case in a mobile phone for example), then it might be that the signal being presented to the ADC circuit already has -40dB of noise present. Noting that dB are on a log scale, we can see that -40dB of noise is far noisy than -74dB. Note the 24dB difference is 8×3dB, and each 3dB corresponds to a doubling of power. Therefore the analog audio signal being presented to the 12-bit ADC would already have 8 times the amount of noise present than what the ADC is going to add.

4.3.2.5 Examples

Compact Disc (CD)

- Stereo, i.e. 2 ADC channels
- $F_s = 44.1\mathrm{kHz}$ per channel
- $b = 16\mathrm{bits}$ resolution per channel

Digital part of Plain Old Telephone Service (POTS)

- Single channel
- $F_s = 8\mathrm{kHz}$
- $b = 8\mathrm{bits}$ resolution

- Analog still used on local loop (to / from the home)