Digital Signal Processing Techniques (An Introduction)

In the previous section we established a link between the digital techniques that we have been using (so far only "running means") and the wider world of filters and so on. What we did there can be derived directly from the general treatment of linear systems and if we are to proceed any further, we must acquaint ourselves with the elements of linear systems in the continuous time domain and in the sampled time domain. This process gets somewhat mathematical - indeed it can sometimes seem almost overwhelmingly mathematical - however bear with me and we shall see the light eventually.

We saw at the end of the last section that the sinc*(x)*[28] form we get for the analysis of a running mean (at least for a running mean over a large number of points so that the transformation from a sum to and integral has some validity) is somewhat different from the frequency response of a filter that we would get from analog design, but it is still a low-pass filter. This implies that we could look at a lot of digital operations on datasets as digital filtering operations and we should look at the question of how the concepts of analog filtering can be carried over into the digital domain. We need to begin with a review of analog concepts relevant to this discussion.

Linear Systems

The fundamental building block of a analog analysis techniques is the concept of a linear system and the impulse response.

A linear system is one in which two rules apply:

1) A given input produces a given output

2) The sum of two inputs produces the sum of the individual outputs. Also known as the principle of linear superposition.

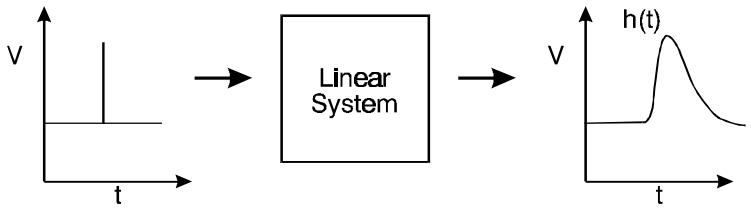
(Drummond's summary: If you kick it, it screams. If you kick it twice as hard, it screams

[28] I'm going to denote sin*x/x* as sinc*(x)* from now on

twice as loud!)

A particular input that is often considered is the impulse function. This is a function which has zero width in time, infinite height, but unit area. It is the Dirac delta function. It is of course impractical to produce delta functions, but approximations are possible which are adequate.

The response of a system to a inpulse function is known as the impulse response and we will designate that as *h(t)*. For a system to be causal it is required that $h(t) = 0$ for $t < 0$. For a system to be stable it is required that $h(t) \rightarrow 0$ as $t \rightarrow \infty$.



Linear System Response

The fundamental concept of linear systems analysis is that the output of a linear system can be predicted for any input if the impulse response is known. In order to understand this concept we realise that the input can be split up into a series of scaled delta functions

$$x(t) \rightarrow \sum_{-\infty}^{\infty} \delta(t - t') x(t') \, dt'$$

and then by appealing to the principle of superposition we can write the output as

$$y(t) = \sum_{t' = -\infty}^{0} x(t + t') \, h(-t')$$

or we can expresss this in integral form as:

$$y(t) = \int_{-\infty}^{0} x(t + t') \, h(-t') \, dt' = \int_{0}^{\infty} x(t - u) \, h(u) \, du$$

where $u = -t'$. Since we also know that $h(u) = 0$ for $u < 0$ for causal systems, we can

extend the integration range to $\pm\infty$ without change and we recognise that the output of a linear system is the convolution of the input with the impulse response. This is a most important result:

The output of a linear system is the convolution of the input and the impulse response. Therefore the impulse response tells you everything there is to know about the linear system.

We introduce a short-hand notation for the convolution integral as

$$y(t) = \int_{-\infty}^{\infty} x(t-u)\ h(u)\ du = x(t) * h(t)$$

Now let us try applying a simple sine wave cosωt to the system.

$$y(t) = \int_{-\infty}^{\infty} \cos\omega(t-u)\ h(u)\ du$$

We could also have chosen sinωt as the input. Since this is so and we can linearly superimpose inputs, let us apply cosωt + $j$sinωt as the input (where $j = \sqrt{(-1)}$). This is an unabashed fiddle to get into complex exponential notation.

$$y(t) = \int_{0}^{\infty} e^{j\omega(t-u)}\ h(u)\ du$$

Since this is a linear system we can write (at least I'm going to tell you I can write) the output as $Y(\omega)\exp(j\omega t)$ where $Y(\omega)$ may be complex.

$$Y(\omega)\ e^{j\omega t} = \int_{0}^{\infty} e^{j\omega t}\ e^{-j\omega u}\ h(u)\ du$$

and since the the term $\exp(j\omega t)$ is a constant of the integration, it can be canceled leading to the result:

$$Y(\omega) = \int_0^{\infty} e^{-j\omega u}\, h(u)\, du$$

If we now back up to our original input - cosωt - and ask what is the output expressed as something multiplied by cosωt, we find that the something is $Y(\omega)$. The modulus of this is what we would call the "frequency response" and the argument is the "phase response". The integral given above is a fourier transform[29]. So the result is:

The frequency response of a linear system is the modulus of the fourier transform of the impulse response and the phase response is its argument.

We introduce a shorthand notation that the fourier transform of $x(t)$ is $X(\omega)$.

Some very important properties of the fourier transform are as follows:

1) The frequency response is always an even function of ω. This is because from the original integral definition $X(\omega) = X^*(-\omega)$ and therefore $|X(\omega)| = |X(-\omega)|$. Similarly the phase term is an odd function of ω.

2) Delaying the time function by a finite time, T, changes only the phase response and that by a simple time delay. (The proof is left as an exercise for the reader)

3) The Fourier transform of the sum of two time functions is the sum of the individual Fourier transforms.

Now let us look at a very useful theorem of fourier transforms which we shall use extensively. Consider the product of two fourier transforms:

$$c(\omega) = \int_{-\infty}^{\infty} a(t)\, \exp(-j\omega t)\, dt \int_{-\infty}^{\infty} b(u)\, \exp(-j\omega u)\, du$$

[29] For those of you who know something about fourier transforms, let me point out that I am not going to normalise anything in what follows. This may not be mathematically correct, but the normalisation doesn't matter much in most cases.

rearranging and substituting $u = v - t$ in the inner integral we obtain:

$$c(\omega) = \int_{-\infty}^{\infty} a(t) \int_{-\infty}^{\infty} b(v-t)\ \exp(-j\omega v)\ dv\ dt$$

Now change the order of the integration

$$c(\omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} a(t)\, b(v-t)\ dt \quad \exp(-j\omega v)\ dv$$

and finally

$$c(\omega) = \int_{-\infty}^{\infty} \left(a(t) * b(t) \right) \quad \exp(-j\omega v)\ dv$$

which states that **the product of two fourier transforms is the fourier transform of the convolution of the two corresponding time functions.**

Thus we can consider two "domains" in which to work, the time domain and the frequency domain. A function can be transformed from time to frequency space by a fourier transform and back by an inverse fourier transform (I didn't prove that, but it is so). Convolution in one domain is the same as multiplication in the other domain.

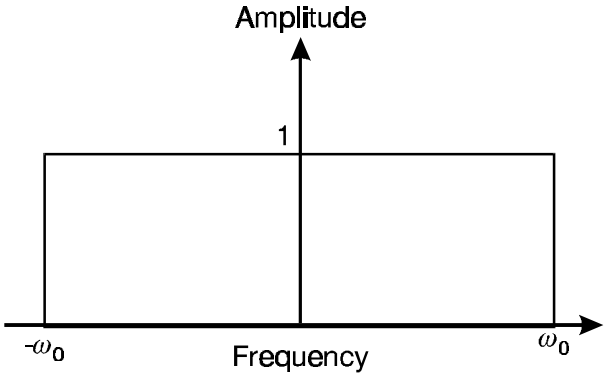
Returning for a moment to the impulse function. Since the output of a linear system is the input function convolved with the impulse function, in frequency space that must be represented by the Frequency spectrum of the input multiplied by the frequency response of the system - which sounds to be about right.

Here's an example. Suppose we have a system whose frequency function is a rectangle of unity height and extent ω_0. This represents a lo-pass filter with a "brick-wall" cut-off at ω_0. Since the frequency representation is required to be even, it must also extend to $-\omega_0$. What is the corresponding impulse function?

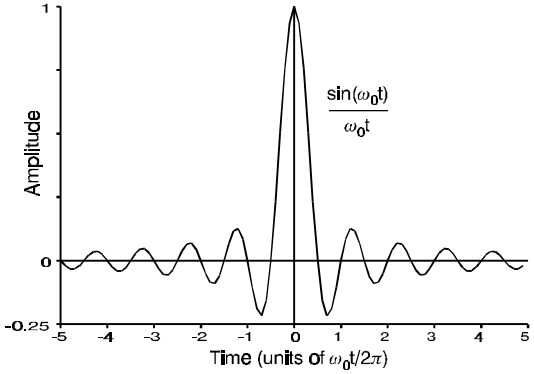
Well let's apply the inverse fourier transform to the function $X(\omega)$

$$x(t) = \int_{-\infty}^{\infty} X(\omega)\ \exp(j\omega t)\ d\omega = \int_{-\omega_0}^{\omega_0} \exp(j\omega t)\ d\omega = \frac{\sin\omega_0 t}{t}$$

which is a sinc*x* function. Notice that if this really is a filter system then it is unrealistic in that it is non-causal since $h(t) \neq 0$ for $t < 0$. We could fix that if the response was exactly zero for $t < -T$ say, by applying a time shift of T to the impulse response. However in this case the response only goes to zero as $t \to \infty$ and this is a problem. This at least partially explains why "brick-wall" filters don't actually exist.



A "Brick-Wall" Filter



"Brick-wall" Filter - Impulse Response

Here's another example. Consider a filter whose passband is so tight that it can be considered to be a delta function in frequency space at frequency ω_0. The impulse function of this filter is:

$$h(t) = \int_{-\infty}^{\infty} \left(\delta(\omega - \omega_0) + \delta(\omega + \omega_0) \right) e^{j\omega t}\ d\omega = 2\cos\omega_0 t$$

which again is non-causal and extends to infinity.

Finally let us consider the transform of a regular array of delta functions at spacing T in the time domain. The result is:

$$X(\omega) \;=\; \int_{-\infty}^{\infty} \sum_{n=-\infty}^{+\infty} \delta(t - nT)\; e^{-j\omega t}\; dt \;=\; \sum_{n=-\infty}^{\infty} e^{jn\omega T}$$

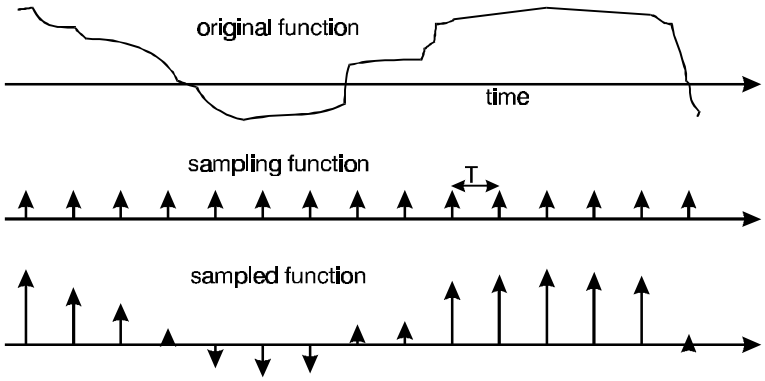
The sum is infinite for values of $\omega = 2k\pi/T$ (k integer) and of (comparatively) negligible size otherwise. This may be approximated to an array of delta functions $\delta(\omega \pm 2k\pi/T)$. Thus an array of delta functions in one domain transforms into an array of delta functions in the other domain.

Sampling of a time function

The action of a simple ADC is to take "spot" values of a time function to produce a sequence of values. If we define

$$x[n] \;=\; x(t)\,\delta(t - nT)$$

where T is the time between samples. We can now write the sampled time series as:



Sampling a Time Function

$$x[t] \;=\; \sum_{n=-\infty}^{\infty} x[n] \;=\; \sum_{n=-\infty}^{\infty} x(t)\,\delta(t - nT)$$

(Notice the notation here where *x[n]* refers to a sample, *x[t]* to the sampled series and *x(t)* to the original continuous time function.)

I can similarly define the discrete version of the impulse function to be:

$$h[n] \;=\; h(t)\,\delta(t - nT)$$

We shall see later that, although this is a useful formal definition, it can be misleading

to consider *h[t]* to be the sampled version of an analog function *h(t)*. However *h[t]* shares with *h(t)* the properties that $h[t] = 0$ for $t < 0$ and $h[t] \rightarrow 0$ for $t \rightarrow \infty$ for causal, stable systems. In real-time processing it is necessary for systems to be causal. However when off-line processing a stored time series, causality is not required.

We can now define the output of a sampled linear system to be the convolution of the input with the impulse function:

$$y[n] = \sum_{m=-\infty}^{\infty} h[m]\ x[n-m]$$

This isn't really an infinite sum because of the restrictions on *h[t]* for causal, stable systems.

If the input function to a discrete linear system is a sampled version of the function $\exp(j\omega t)$.

$$x[n] = e^{j\omega nT}$$

Using the principle of convolution the the output will be:

$$\sum_{m=-\infty}^{\infty} e^{j\omega(n-m)T}\ h[m] = e^{j\omega nT} \sum_{m=-\infty}^{\infty} e^{-j\omega mT}\ h[m]$$

which is the discrete analogue of the convolution of the input and the impulse function. This can be written as:

$$Y[n] = H[\omega]\, e^{j\omega nT}$$

Where we have introduced the discrete analogue of the fourier transform. However now let us use an input

$$x[n] = \exp\left(j(\omega + \frac{2\pi}{T})nT \right)$$

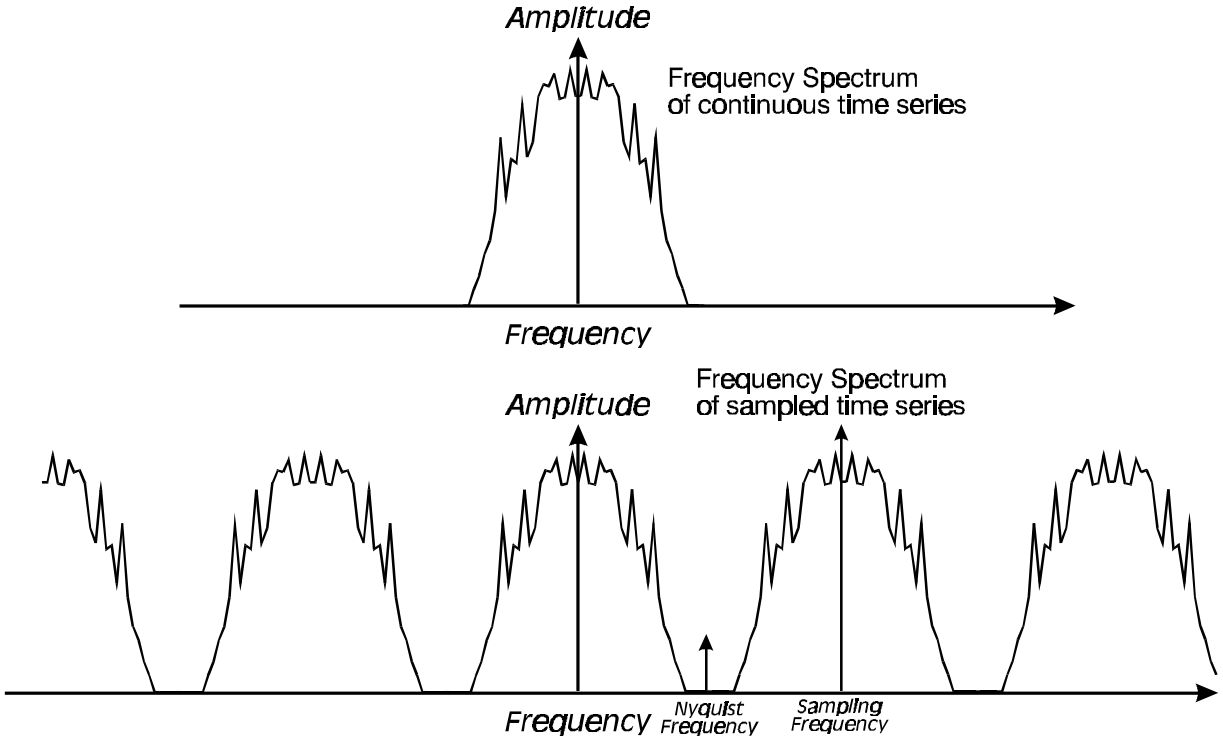
This produces the output:

$$\sum_{m=-\infty}^{\infty} \exp(j(\omega+2\pi/T)(n-m)T)\ h[m]$$

$$= \exp(j(\omega+2\pi/T)nT) \sum_{m=-\infty}^{\infty} \exp(-j(\omega+2\pi/T)mT)\ h[m]$$

However we note that $\exp(2n\pi) = 1$ for all integer n and therefore this becomes:

$$Y[n] = H[\omega] \exp(j(\omega+2\pi/T)nT) = H[\omega]\, e^{j\omega nT}$$



Comparison of Continuous Function and Its Sampled Form

In other words we cannot distinguish at the output between inputs which are spaced $2\pi/T$ away from each other in angular frequency and the frequency response is also the same. In fact since we can repetitively apply this criterion we find that the "aliasing" of frequencies (being unable to distinguish at the output between various input frequencies) and repetition

of the system response occurs at all integer multiples of $2\pi/T$. Since the frequency response is an even function, this further implies that the frequency response looks as shown in the accompanying diagram - even about the origin and repeating with a period of $2\pi/T$ in angular frequency, $1/T$ in linear frequency.

Another way of deriving this would be to say that the sampled time series is the product of the continuous time function and an array of delta functions. Therefore the frequency domain representation must be the convolution of the transforms of the two functions - the transform of the original time function and the transform of a regular array of delta functions, which is itself a regular array of delta functions in the frequency domain.

If any input above π/T in angular frequency is passed through the system, then it will also appear as another frequency in the range $0 \rightarrow \pi/T$. In linear frequency terms this states that the system frequencies must be limited to $f < 1/(2T)$ or that input frequencies must be sampled at least twice per cycle.

This is the sampling or Nyquist criterion which states:

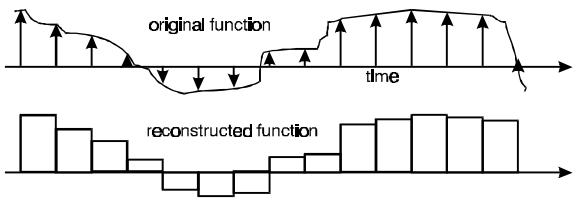
The highest frequency present in the system must be sampled at least twice per cycle.

The limiting frequency of $1/2T$ where T is the sampling time is often called the Nyquist frequency.

The other properties of the fourier transform mentioned above for the continuous case are unchanged: The transforms are even, additive and a time delay in the time domain causes only a difference in the phase term. It is also true that convultion in one domain is the same as multiplication in the other domain.

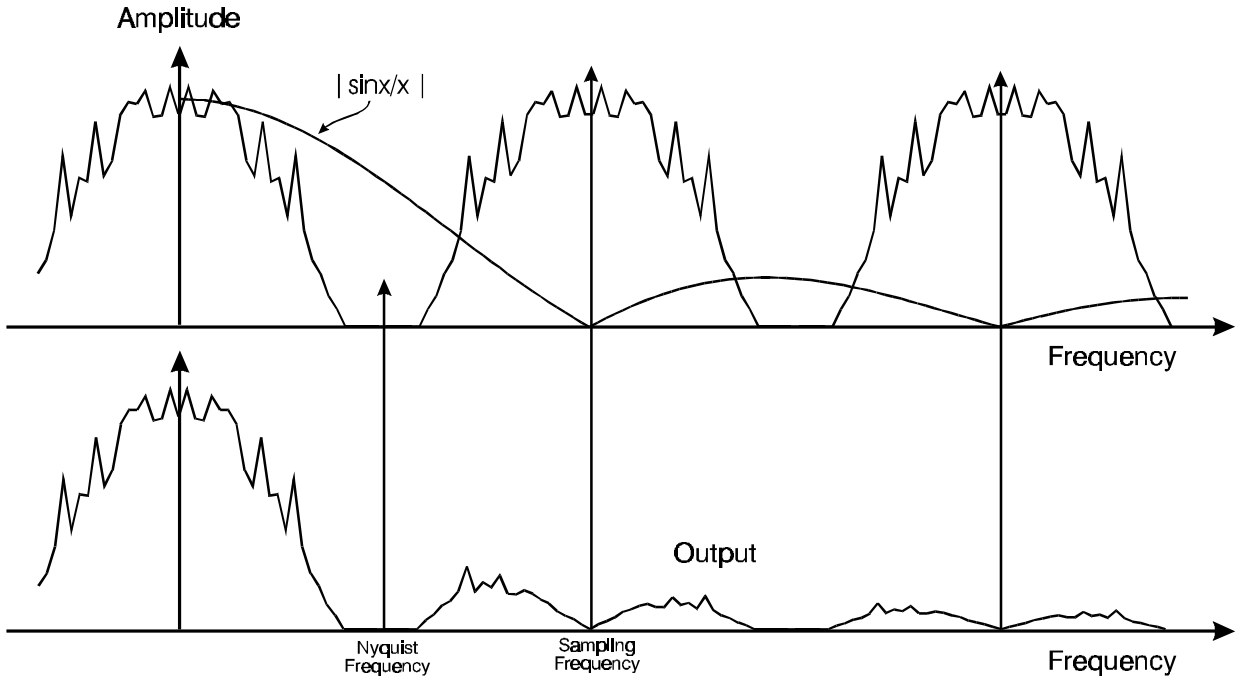
Interpolation

Now let us consider some more serious issues. Suppose that we take a sampled series and pass it through a DAC to retrieve a continuous function again. We can assume for a moment that the DAC is perfect, "glitchless" and infinitely fast. The output of the DAC consists of a series of levels which change at a



Reconstructed Function After DAC

rate equal to the series sampling rate. In fact with a bit of thought we realise that the output of the DAC can be thought of as the sampled series convolved with a square pulse of unit height whose width is the sampling time. Now convolution in the time domain is equivalent to multiplication in the frequency domain so the frequency content of the output is the product of the fourier transforms of the two functions. The time series has a fourier transform which is even and repeats with a frequency of the inverse of the sampling rate. The pulse has a transform of sinc(π/T) with a zero at $1/T$ in linear frequency. The result is as shown in the diagram.



Effect of DAC on Sampled Waveform

This has significant implications for a DAC signal as the output will contain aliased frequencies since the cut-off is not at all sharp after the Nyquist frequency. There is also a problem in that the curve also attentuates higher frequencies in the non-aliased part of the spectrum - it acts as a filter.

Now let us consider a fairly radical step. We will take the time series and double its rate - ie halve the sampling time. We will need to insert a number between every pair of samples to do this and temporarily we will select the number zero. We will first need to find the frequency spectrum of the new time series.

We can do this conceptually by multiplying the time series by another time series which is an array of delta functions of interval $T/2$. This has a fourier transform of an array of delta functions at intervals of $2/T$ in linear frequency and multiplication in the time domain implies convolution in the frequency domain. The resulting frequency plot is exactly the same as the fourier transform of the original time series - but the Nyquist and sampling frequencies are twice as high.

Now if we pass this time series through the DAC the result will be worse than before because the convolving pulse will result in a first zero at $2/T$ which passes more of the aliased stuff through. We need to do something about those zeros.

Suppose we pass the time series through a "filter" which replaces each zero with the previous value. This is equivalent (apart from a time delay which we can ignore) to the impulse function:

$$h[0] = 1$$
$$h[1] = 1$$

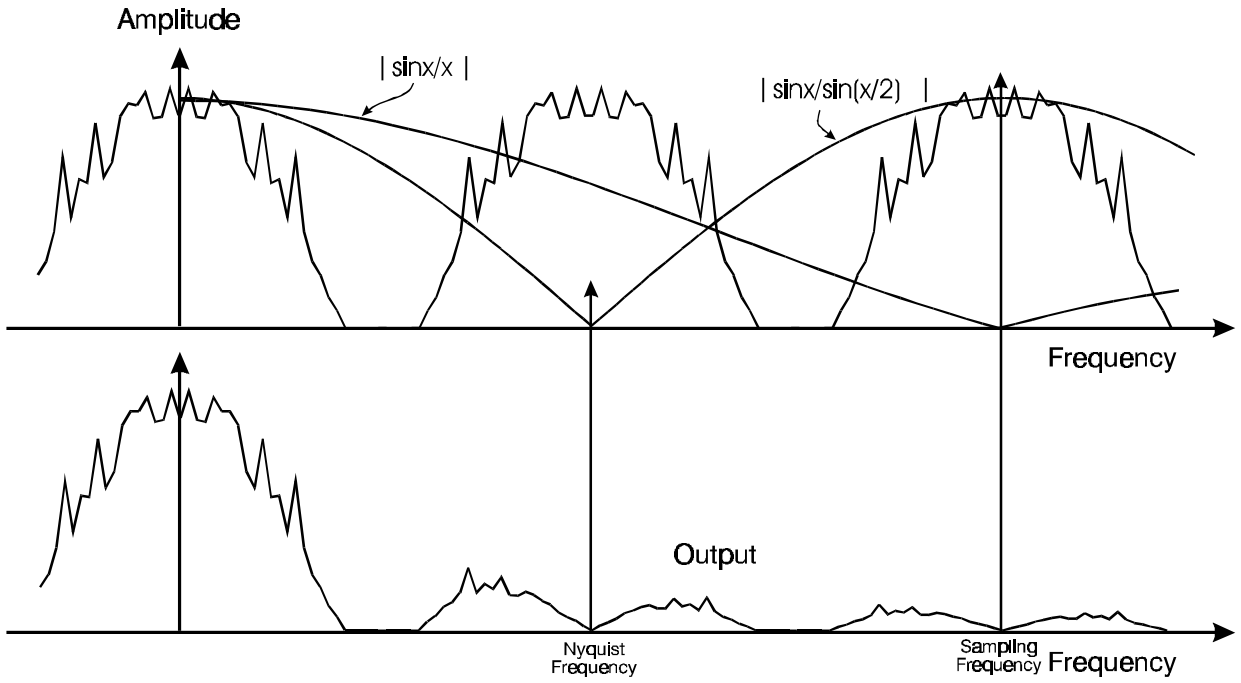
with all other terms 0. This looks like a pulse function for which the fourier transform would be sinc$x$ - exactly the same as the original pulse. However this is a sampled system and we need to get the sampled for of the fourier transform. The fourier transform in this case is:

$$H[\omega] = \sum_{n=-\infty}^{\infty} h[n]\, e^{j\omega nT/2} = 1 + e^{j\omega T/2}$$

The modulus of this is:

$$|H[\omega]| = \frac{\sin(\omega T/2)}{\sin(\omega T/4)}$$

which is even in ω and repeats with linear frequency $2/T$ - as you would expect



We finally have to account for the fact that the DAC is still going to hold the value between samples which gives a sinc*x* response appropriate to a sampling rate of $T/2$. If we multiply the two "filters" together we find that the effect of both operations is actually to return the system to the performance of the the original series sampled at $1/T$ - a long way around to get nowhere!

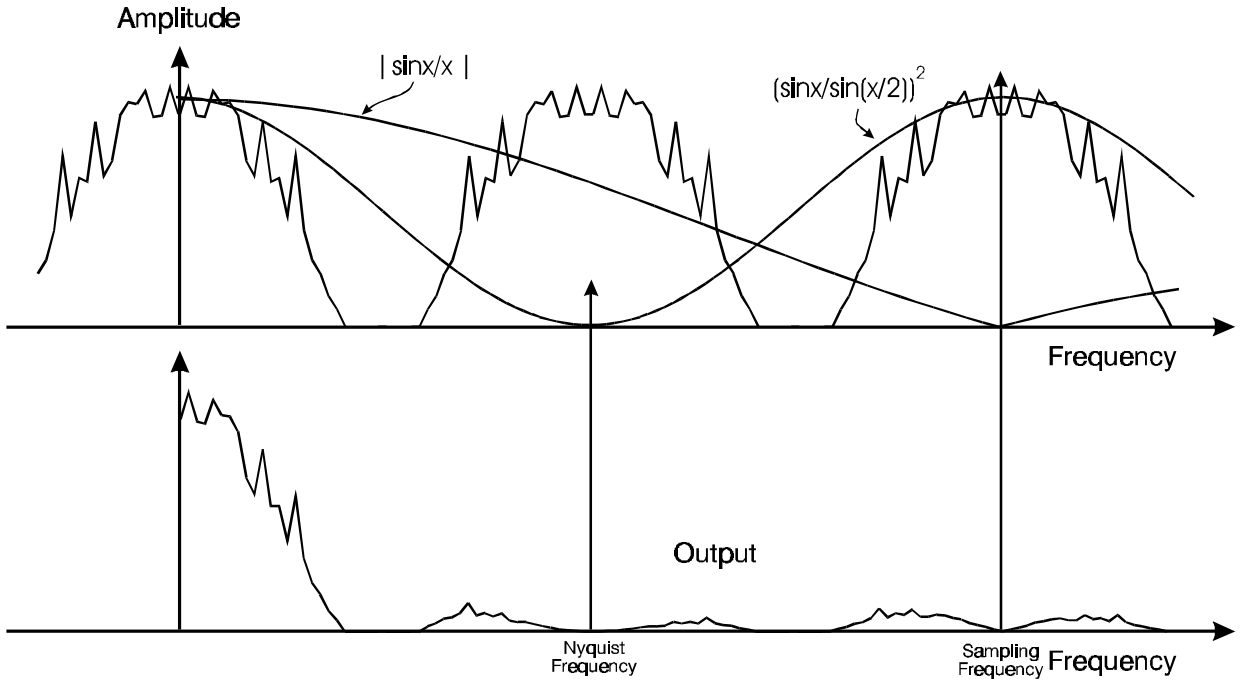
$$\frac{\sin(\omega T/2)}{\sin(\omega T/4)} \; \frac{\sin(\omega T/4)}{\omega}$$

However suppose that instead of replacing each pulse by the previous one, we linearly interpolate between the pulses. This is equivalent (apart from a time delay again) to the impulse function

$$h[0] = 0.5$$
$$h[1] = 1.0$$
$$h[2] = 0.5$$

With all other terms zero. This is a triangle function and has a fourier transform of

$$H(\omega) = \left(\frac{\sin(\omega T/2)}{\sin(\omega T/4)} \right)^2$$



Interpolation Filter compared with "Hold" filter

If we compare the result graphically with the previous filter (replace zeros by last value) we can see a distinct improvement in the suppression of the higher frequencies - we are now improving things over the simple circuit.

Now all the mathematics here has been done for simply one sample inserted between each sample of the function. It is evidently possible to add more samples - in which case there is more scope for filtering the output by the form of the reconstruction of the intermediate samples[30].

[30] This is sometimes called "oversampling" in the audio world - we have been discussing 2x oversampling.