# Digital Filters

When looking at digital filters, there are two classifications used

* Infinite Impulse Response (IIR)
* Finite Impulse Response (FIR)

## Ideal Filter Response

The ideal low-pass filter frequency response and its time domain impulse response are shown below

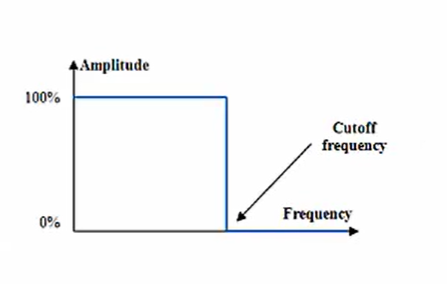


Figure : Ideal Frequency Response

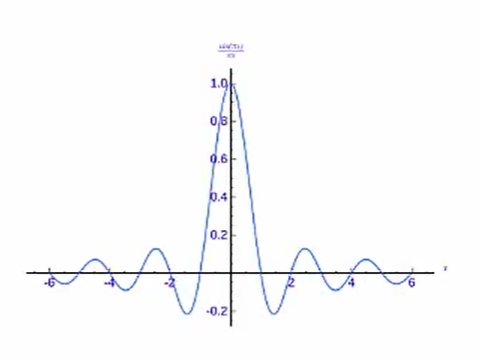


Figure : Time Domain Impulse Response

Note that while we could sample and use this impulse response, it is both infinite and non-causal, meaning it cannot be used for real time, on-line filtering.

## Infinite Impulse Response (IIR)

IIR refers to a system/filter which has poles in its transfer functions. All analogue filters will fall into this category since it is not possible to synthesise an analogue filter without poles. The presence of poles means an IIR filter could potentially become unstable.

In general, IIR filters offer better frequency domain performance than FIR filters. However, they cannot be fully realised via the FFT due to the infinite in infinite impulse response. This problem can be overcome by choosing a finite portion of the infinite impulse response to look at.

### Techniques for IIR Design

By far the most common technique for IIR filter design is to perform the design using the analogue domain. This may seem strange since we are talking about designing digital filters.

Because of the amount of information available for analogue filters, as well as the number of techniques for making the filters means it is pointless to attempt to reproduce this work for the digital domain.

There is also the fact that there are relatively easy methods for transforming an analogue filter’s transfer function (TF) to a digital TF (Tustin’s bilinear transform) makes IIR design relatively simple when starting with an analogue design.

This means the IIR design process will be to find a digital transfer function whose frequency response most closely approximates that of the analogue prototype.

## Finite Impulse Response (FIR)

FIR refers to a system without poles. In general, FIRs offer better time domain performance (e.g. fewer calculations, less memory required) than IIR. In a real-world setting, this typically translates to lower system cost.

Since there are no poles in a FIR implementation, the filter will never be unstable.

### FIR Filter Design

FIR filters have no analogue counterpart. As previously noted, the FIR filter has a transfer function with no poles. This equates to no feedback in the filter. This leads to certain advantages:

* FIR filters are guaranteed to be stable due to the lack of feedback
* It is very simple to make the phase response linear
* Efficient to computer in real time via the FFT

There are several methods available for FIR design

* Window/Truncation of an ideal filter
* Frequency domain sampling and iFFT
* Least squares approximation to ideal filter response

## The Bilinear Transform

The bilinear transform, is defined as:

This gives us a direct method of substitution to obtain a digital TF from the analogue one (which would be a function of the complex frequency *s.*

However, the bilinear transform has the property that it maps the entire imaginary axis of the s-plane to the unit circle of the z-plane. This means we have an infinitely long line mapped to a finite length curve. As such, the imaginary axis must be shortened to fit. This results in what we call frequency wrapping. The further we move around the unit circle after using the bilinear transform, the bigger the difference is between the digital frequency at that point and its “corresponding” analogue frequency.

### Frequency Wrapping

The relationship between analogue frequency and digital under the bilinear transform is as follows:

We can see that the distortion gets more pronounced the closer the analogue frequency gets to the Nyquist rate .

So how do we get around this? What we can do is pre-warp our frequency constraints before we design the analogue filter, so that we end up with the desired performance in the digital filter.

Pre-warping is accomplished by inverting the frequency warping relation shown above.

Note that because we are designing the filter after we pre-wrap, we can choose an arbitrary sampling period here (which is independent of the actual sampling period of the implementation of the filter). Thus, with judicious choice of the sampling rate, the pre-warping formula becomes

With the bilinear transform and pre-warping formula in hand, we are ready to design an IIR filter. The process is as follows:

* Specify the filter performance criteria as required
* Pre-warp corner frequency/frequencies
* Design an analogue filter which meets the performance at the pre-warp frequencies
* Substitute for *s* in the transfer function of the analogue filter using the bilinear transformation.
* Use the resulting digital transfer function to implement the filter in software/hardware.