Lecture 5 - Digital Filters

5.1 Digital filtering

Digital filtering can be implemented either in hardware or software; in the first case, the numerical processor is either a special-purpose chip or it is assembled out of a set of digital integrated circuits which provide the essential building blocks of a digital filtering operation – storage, delay, addition/subtraction and multiplication by constants. On the other hand, a general-purpose mini-or micro-computer can also be programmed as a digital filter, in which case the numerical processor is the computer's CPU and memory.

Filtering may be applied to signals which are then transformed back to the analogue domain using a *digital to analogue convertor* or worked with entirely in the digital domain (e.g. biomedical signals are digitised, filtered and analysed to detect some abnormal activity).

5.1.1 Reasons for using a digital rather than an analogue filter

- The numerical processor can easily be (re-)programmed to implement a number of different filters.
- The accuracy of a digital filter is dependent only on the round-off error in the arithmetic. This has two advantages:
  - the accuracy is *predictable* and hence the performance of the digital signal processing algorithm is known *a priori*.
  - The round-off error can be minimized with appropriate design techniques and hence digital filters can meet very tight specifications on magnitude and phase characteristics (which would be almost impossible to achieve with analogue filters because of component tolerances and circuit noise).

- The widespread use of mini- and micro-computers in engineering has greatly increased the number of digital signals recorded and processed. Power supply and temperature variations have no effect on a programme stored in a computer.
- Digital circuits have a much higher noise immunity than analogue circuits.

5.1.2 The sampling process

This is performed by an *analogue to digital converter* (ADC) in which the continuous function $f(t)$ is replaced by a "discrete function" $f[k]$, which is defined only at $t = kT$, with $k = 0, 1, 2$. We thence only need consider the digitised sample set $f[k]$ and the sample interval T.

Aliasing

Consider $f(t) = \cos(\frac{\pi}{2}\frac{t}{T})$ (one cycle every 4 samples) and also $f(t) = \cos(\frac{3\pi}{2}\frac{t}{T})$ (3 cycles every 4 samples) as shown in Fig. 5.1. Note that the resultant samples are the same. This result is referred to as aliasing. The most popular solution to this problem is to precede the digital filter by an *analogue* low-pass filter to ensure that any frequency above $1/2T$ Hz is adequately suppressed. (This LPF is known as an "anti-aliasing" filter and is shown to the left of the A/D converter on the block diagram of page 46). Without an anti-aliasing filter, $0.3/T$ Hz at the output could have been produced either by $0.3/T$ Hz or $0.7/T$ Hz at the input. With the anti-aliasing filter, any signal at $0.7/T$ Hz will be suppressed prior to A/D conversion, i.e. $0.3/T$ coming out must represent $0.3/T$ going in. Since we cannot build perfect anti-aliasing filters, however, digital filters will normally have their passband well below $1/2T$ Hz.

5.1.3 Reconstruction of output waveform

The data at the output of the D/A converter must be interpolated in order to reconstruct the filtered signal. Usually this takes the form of step or one-point interpolation (in which the value at the sample time is assumed to be the value of the function until the next sample time), followed by analogue low-pass filtering as shown in Fig. 5.2. The output low-pass filter is sometimes known as a recovery filter.

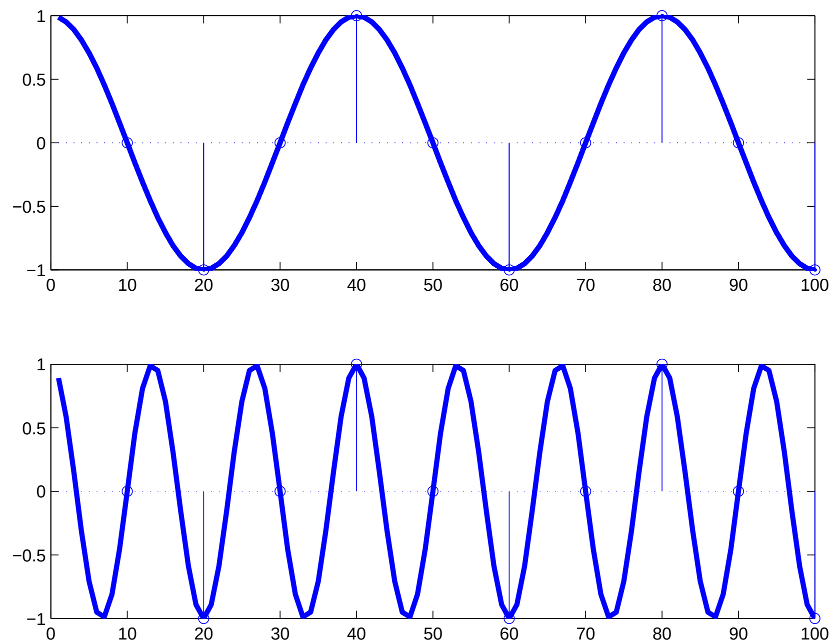

Figure 5.1: *Aliasing.*

5.2 Introduction to the principles of digital filtering

We can see that the numerical processing is at the heart of the digital filtering process. How can the arithmetic manipulation of a sequence of numbers produce a "filtered" version of that sequence? Consider the noisy signal of figure 5.3, together with its sampled version:

One way to reduce the noise might be to try and *smooth* the data. For example, we could try a polynomial fit using a least-squares criterion. If we choose, say, to fit a parabola to every group of 5 points in the sequence, then, for every point, we will make a parabolic approximation to that point using the value of the sample at that point together with the values of the 4 nearest samples (this forms a *parabolic filter*), as in Fig. 5.4

$$p[k] = s_0 + ks_1 + k^2 s_2$$

where $p[k]$ = value of parabola at each of the 5 possible values of $k = \{-2, -1, 0, 1, 2\}$ and s_0, s_1, s_2 are the variables used to fit each of the parabolae to 5 input data points.

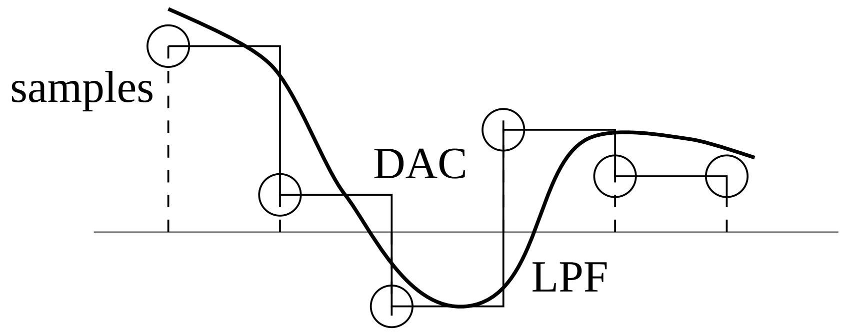

Figure 5.2: *Reconstruction using a DAC and LPF.*

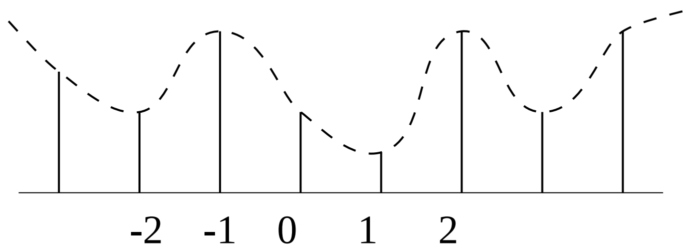

Figure 5.3: *Noisy data.*

We obtain a fit by finding a parabola (coefficients s_0, s_1 and s_2) which best approximates the 5 data points as measured by the least-squares error E:

$$E(s_0, s_1, s_2) = \sum_{k=-2}^{2} (x[k] - [s_0 + ks_1 + k^2 s_2])^2$$

Minimizing the least-squares error gives:

$$\frac{\partial E}{\partial s_0} = 0, \quad \frac{\partial E}{\partial s_1} = 0, \quad \text{and } \frac{\partial E}{\partial s_2} = 0$$

and thus:

$$5s_0 + 10s_2 = \sum_{k=-2}^{k=2} x[k]$$

$$10s_1 = \sum_{k=-2}^{k=2} kx[k]$$

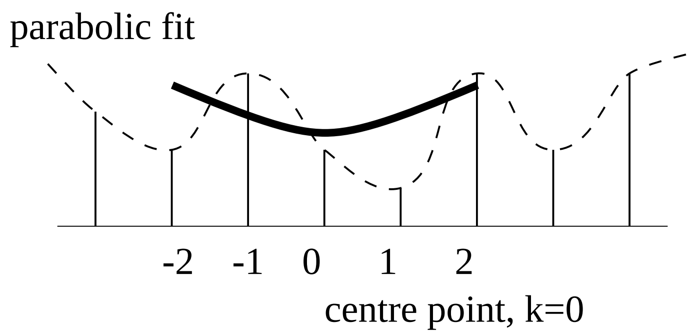

Figure 5.4: *Parabolic fit.*

$$10s_0 + 34s_2 = \sum_{k=-2}^{k=2} k^2 x[k]$$

which therefore gives:

$$s_0 = \frac{1}{35}(-3x[-2] + 12x[-1] + 17x[0] + 12x[1] - 3x[2])$$

$$s_1 = \frac{1}{10}(-2x[-2] - x[-1] + x[1] + 2x[2])$$

$$s_2 = \frac{1}{14}(2x[-2] - x[-1] - 2x[0] - x[1] + 2x[2])$$

The centre point of the parabola is given by:

$$p[k] \mid_{k=0} = s_0 + ks_1 + k^2 s_2 \mid_{k=0} = s_0$$

Thus, the parabola coefficient s_0 given above is the output sequence number calculated from a set of 5 input sequences points. The output sequence so obtained is similar to the input sequence, but with less noise (i.e. low-pass filtered) because the parabolic filtering provides a smoothed approximation to each set of five data points in the sequence. Fig. 5.5 shows this filtering effect. The magnitude response (which we will re-consider later) for the 5-point parabolic filter is shown below in Fig. 5.6.

The filter which has just been described is an example of a *non- recursive* digital filter, which are defined by the following relationship (known as a *difference*

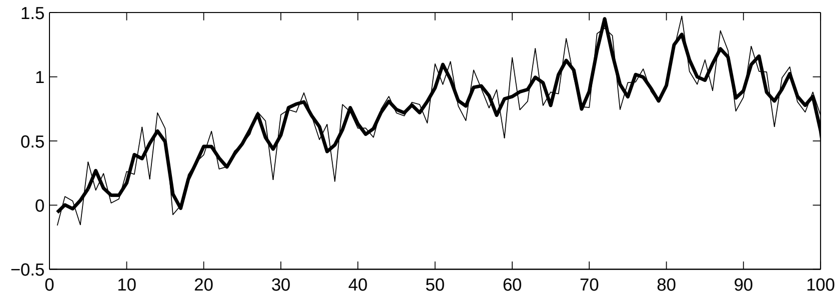

Figure 5.5: *Noisy data (thin line) and 5-point parabolic filtered (thick line).*

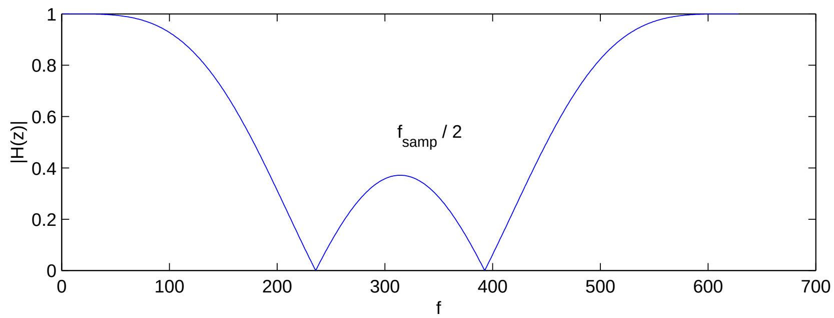

Figure 5.6: *Frequency response of 5-point parabolic filter.*

equation):

$$y[k] = \sum_{i=0}^{N} a_i x[k-i]$$

where the a_i coefficients determine the filter characteristics. The difference equation for the 5-point smoothing filter, therefore, is:

$$y[k] = \frac{1}{35}(-3x[k+2] + 12x[k+1] + 17x[k] + 12x[k-1] - 3x[k-2])$$

This is a *non-causal* filter since a given output value $y[k]$ depends not only on previous inputs, but also on the current input $x[k]$, the input $x[k+1]$ and the input $x[k+2]$. The problem is solved by delaying the calculation of the output value $y[k]$ (the centre point of the parabola) until all the 5 input values have been sampled (i.e. a delay of $2T$ where T = sampling period), ie:

$$y[k] = \frac{1}{35}(-3x[k] + 12x[k-1] + 17x[k-2] + 12x[k-3] - 3x[k-4])$$

It is of importance to note that the equation $y[k] = \sum a_i x[k-i]$ represents a *discrete convolution* of the input data with the filter coefficients; hence these coefficients constitute the *impulse response* of the filter.

Proof:

Let $x[k] = 0$, except at $k = 0$, where $x(0) = 1$. Then $y[k] = \sum_i a_i x[k-i] = a_k x(0)$ (all terms zero except when $i = k$). This is equal to a_k since $x(0) = 1$. Therefore $y(0) = a_0$; $y(1) = a_1$; etc .... As there is a finite number of a's, the impulse response is finite. For this reason, non-recursive filters are also called **Finite-Impulse Response** (FIR) filters.

As we will see, we may also formulate a digital filter as a recursive filter; in which, the output $y[k]$ is also a function of previous outputs:

$$y[k] = \sum_{i=0}^{N} a_i x[k-i] + \sum_{i=1}^{M} b_i y[k-i]$$

Before we can describe methods for the design of both types of filter, we need to review the concept of the z-transform.

5.3 The z-transform

The z-transform is important in digital filtering because it describes the sampling process and plays a role in the digital domain similar to that of the Laplace transform in analogue filtering.

The Laplace transform of a unit impulse occurring at time $t = kT$ is e^{-kTs}. Consider the discrete function $f[k]$ to be a succession of impulses, for example of area $f(0)$ occurring at $t = 0$, $f(1)$ occurring at $t = T$, etc .... The Laplace transform of the whole sequence would be:

$$F_d(s) = f(0) + f(1)e^{-Ts} + f(2)e^{-2Ts} + \ldots + f[k]e^{-kTs}$$

The suffix d denotes the transform of the *discrete* sequence, not of the continuous $f(t)$.

Let us replace e^{Ts} by a new variable z, and rename $F_d(s)$ as $F(z)$:

$$F(z) = f(0) + f(1)z^{-1} + f(2)z^{-2} + \ldots f[k]z^{-k}$$

For many functions, the infinite series can be represented in "closed form", in general as the ratio of two polynomials in z^{-1}.

5.3.1 Examples of the z-transform

- **step function** $f[k]$ = 0 for $k < 0$, $f[k]$ = 1 for $k \geq 1$

$$F(z) = 1 + z^{-1} + z^{-2} + \ldots + z^{-k} + \ldots$$

$$F(z) = \frac{1}{1 - z^{-1}}$$

by summation of a geometric progression[6].

- **Decaying exponential**

$$f(t) = e^{-\alpha t} \longrightarrow f[k] = e^{-\alpha kT}$$

$$F(z) = 1 + e^{-\alpha T}z^{-1} + e^{-\alpha 2T}z^{-2} + \ldots + e^{-\alpha kT}z^{-k} + \ldots$$

$$F(z) = \frac{1}{1 - e^{-\alpha T}z^{-1}}$$

- **Sinewaves**

$$f(t) = \cos \omega t \longrightarrow f[k] = \cos k\omega T = \frac{e^{jk\omega T} - e^{-jk\omega T}}{2}$$

$$F(z) = \frac{1}{2}(\frac{1}{1 - e^{j\omega T}z^{-1}} + \frac{1}{1 - e^{-j\omega T}z^{-1}})$$

$$F(z) = \frac{1 - \cos \omega T z^{-1}}{1 - 2\cos \omega T z^{-1} + z^{-2}}$$

[6]strictly valid only if $\mid z^{-1} \mid < 1$

5.3.2 The Pulse Transfer Function

This is the name for (z-transform of output)/(z-transform of input).

Let the impulse response, for example of an FIR filter, be a_0 at $t = 0$, a_1 at $t = T$, $\ldots a_i$ at $t = iT$ with $i = 0$ to N.

Let $G(z)$ be the z-transform of this sequence:

$$G(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_i z^{-i} + \ldots a_N z^{-N}$$

Let $X(z)$ be an input:

$$X(z) = x[0] + x[1]z^{-1} + x[2]z^{-2} + \ldots + x[k]z^{-k} + \ldots$$

The product $G(z)X(z)$ is:

$$G(z)X(z) = (a_0 + a_1 z^{-1} + \ldots a_i z^{-i} + \ldots a_N z^{-N})(x[0] + x[1]z^{-1} + \ldots x[k]z^{-k})$$

in which the coefficient of z^{-k} is:

$$a_0 x[k] + a_1 x[k-1] + \ldots a_i x[k-i] + \ldots a_N x[k-N]$$

This is nothing else than the value of the output sample at $t = kT$. Hence the whole sequence is the z-transform of the output, say $Y(z)$, where $Y(z) = G(z)X(z)$. Hence the pulse transfer function, $G(z)$, is the z-transform of the impulse response.

For non-recursive filters:

$$G(z) = \sum_{i=0}^{N} a_i z^{-i}$$

For recursive filters:

$$Y(z) = \sum_{i=0}^{N} a_i z^{-i} X(z) + \sum_{i=1}^{M} b_i z^{-i} Y(z)$$

$$G(z) = \frac{Y(z)}{X(z)} = \frac{\sum a_i z^{-i}}{1 - \sum b_i z^{-i}}$$

5.3.3 z-plane pole-zero plot

Let $z = e^{sT}$, where T = sampling period. Since $s = \sigma + j\omega$, we have:

$$z = e^{\sigma T} e^{j\omega T}$$

If $\sigma = 0$, then $\mid z \mid = 1$ and $z = e^{j\omega T} = \cos\omega T + j\sin\omega T$, i.e. the equation of a circle of unit radius (the *unit circle*) in the z-plane.

Thus, the imaginary axis in the s-plane ($\sigma = 0$) maps onto the unit circle in the z-plane and the left half of the s-plane ($\sigma < 0$) onto the *interior* of the unit circle.

We know that all the poles of $G(s)$ must be in the left half of the s-plane for a continuous filter to be stable. We can therefore state the equivalent rule for stability in the z-plane:

For stability all poles in the z-plane must be inside the unit circle.

5.4 Frequency response of a digital filter

This can be obtained by evaluating the (pulse) transfer function on the unit circle (i.e. $z = e^{j\omega T}$).

Proof
Consider the general filter

$$y[k] = \sum_{i=0}^{\infty} a_i x[k-i]$$

NB: A recursive type can always be expressed as an infinite sum by dividing out:

$$\text{eg., for} \quad G(z) = \frac{a_0}{1 - b_1 z^{-1}}, \quad \text{we have} \quad y[k] = \sum_{i=0}^{\infty} a_0 . b_1^i x[k-i]$$

Let input before sampling be $\cos(\omega t + \theta)$, sampled at $t = 0, T, \ldots, kT$. Therefore $x[k] = \cos(\omega kT + \theta) = \frac{1}{2}\{e^{j(\omega kT+\theta)} + e^{-j(\omega kT+\theta)}\}$

$$\text{ie.} y[k] = \frac{1}{2}\sum_{i=0}^{\infty} a_i e^{j\{\omega[k-i]T+\theta\}} + \frac{1}{2}\sum_{i=0}^{\infty} a_i e^{-j\{\omega[k-i]T+\theta\}}$$

$$= \frac{1}{2} e^{j(\omega kT+\theta)} \sum_{i=0}^{\infty} a_i e^{-j\omega iT} + \frac{1}{2} e^{-j(\omega kT+\theta)} \sum_{i=0}^{\infty} a_i e^{j\omega iT}$$

$$\text{Now} \sum_{i=0}^{\infty} a_i e^{-j\omega iT} = \sum_{i=0}^{\infty} a_i (e^{j\omega T})^{-i}$$

$$\text{But} G(z) \text{ for this filter is } \quad \sum_{i=0}^{\infty} a_i z^{-i}$$

$$\text{and so} \quad \sum_{i-0}^{\infty} a_i e^{-j\omega iT} = G(z)_z = e^{j\omega T}$$

Let $G(z)_z = e^{j\omega T} = Ae^{j\phi}$.
Then

$$\sum_{i=0}^{\infty} a_i e^{j\omega iT} = Ae^{-j\phi} \quad \text{(complex conjugate)}$$

Hence $y[k] = \frac{1}{2} e^{j(\omega kT+\theta)} Ae^{j\phi} + \frac{1}{2} e^{-j(\omega kT+\theta)} Ae^{-j\phi}$

$$\text{or} \quad y[k] = A\cos(\omega kT + \theta + \phi) \quad \text{when } x[k] = \cos(\omega kT + \theta)$$

Thus A and ϕ represent the gain and phase of the frequency response. i.e. the frequency response (as a complex quantity) is

$$G(z)|_{z=e^{j\omega T}}$$

Example

Consider the 5-point parabolic filter.

$$y[k] = \frac{1}{35}(-3x[k] + 12x[k-1] + 17x[k-2] + 12x[k-3] - 3x[k-4])$$

hence

$$Y(z) = \frac{1}{35}(-3 + 12z^{-1} + 17z^{-2} + 12z^{-3} - 3z^{-4})X(z)$$

thus

$$G(z)_{z=e^{j\omega T}} = \frac{1}{35}(-3 + 12e^{-j\omega T} + 17e^{-2j\omega T} + 12e^{-3j\omega T} - 3e^{-4j\omega T})$$

$$G(e^{j\omega T}) = \frac{1}{35}e^{-2j\omega T}(17 + 24\cos\omega T - 6\cos 2\omega T)$$

Therefore $\quad |G(e^{j\omega T})| = \frac{1}{35}|17 + 24\cos\omega T - 6\cos 2\omega T|$

When $\omega T = 0$, $|G(e^{j\omega T})| = 1$
and $|G(e^{j\omega T})| = 0.707$ when $\omega T \approx 0.48\pi$, i.e. $\frac{f}{f_s} = 0.24$
$\angle G(e^{j\omega T}) = -2\omega T$

i.e. linear phase (true for any FIR filter with palindromic coefficients); all frequencies delayed by $2T$ (as expected!)