Lecture 2 - The Fourier Domain & Digital Filters

2.1 The Fourier transform

Consider again the 1-D case of a signal $f(x)$, the FT is defined as

$$F(u) = \int_{-\infty}^{+\infty} f(x) \exp[-i2\pi ux] dx$$

and the inverse as

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{i2\pi ux} du$$

which form a *Fourier transform pair*. We note that the FT is, in general, a complex function of the form $F(u) = \mathbb{R}(u) + i\mathbb{I}(u)$. We call $|F(u)|$ the *Fourier spectrum* of $f(x)$ and $\phi(u) = \tan^{-1}[\mathbb{I}(u)/\mathbb{R}(u)]$ the *phase spectrum*.

2.1.1 2-D Fourier transform

There is no inherent change in theory for the 2-dimensional case, where $f(x, y)$ exists, so the FT, $F(u, v)$ is given as

$$F(u, v) = \int \int_{-\infty}^{+\infty} f(x, y) \exp[-i2\pi(ux + vy)] dxdy$$

and the inverse as

$$f(x, y) = \int \int_{-\infty}^{+\infty} F(u, v) \exp[i2\pi(ux + vy)] dudv$$

We note that the above are *separable*

2.1.2 Some basic theorems

Here are some basic (and useful) theorems related to the FT. They are shown for a 1-D system, for ease of reading and notation, and directly translate into higher dimensions as above.

- *Similarity theorem* : if $f(x) \to F(u)$ then $f(ax) \to \frac{1}{|a|}F(u/a)$
- *Addition theorem* : if $f(x), g(x) \to F(u), G(u)$ then $af(x)+bg(x) \to aF(u)+bG(u)$
- *Shift or twist theorem* : if $f(x) \to F(u)$ then $f(x-a) \to \exp[-i2\pi ua]F(u)$
- *Convolution theorem* : if

  $$f(x) * g(x) = \int f(\tau)g(x-\tau)d\tau$$

  then $FT[f(x) * g(x)] = F(u)G(u)$. Note that this is of great use in filtering
- *Power theorem* :

  $$\int |f(x)|^2 dx = \int |F(u)|^2 du$$

  i.e. a statement about conservation of energy.
- *Derivative theorem* : if $f(x) \to F(u)$ then $f'(x) = d/dx[f(x)] \to iuF(u)$

2.1.3 The discrete FT (DFT)

We sample the continuous (start with 1-D) function, $f(x)$, at M points spaced Δx apart. We now describe the function as

$$f(x) = f(x_o + x\Delta x)$$

where x now describes an *index*, with this transformation, u the Fourier varable paired to x is discretised into M points. We thus obtain :

$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)\exp[-i2\pi ux/M]$$

and

$$f(x) = \sum_{u=0}^{M-1} F(u)\exp[i2\pi ux/M]$$

and, for 2-D systems

$$F(u,v) = \frac{1}{M}\frac{1}{N}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\exp[-i2\pi(ux/M + vy/N)]$$

and

$$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)\exp[i2\pi(ux/M + vy/N)]$$

if y is sampled evenly at N sample points.

The sampling in the space domain, $\Delta x, \Delta y$ corresponds to a sampling in the ‘frequency’ domain of

$$\Delta u = \frac{1}{M\Delta x}$$
$$\Delta v = \frac{1}{M\Delta y}$$

2.1.4 Some useful results using the DFT

- The total width of the samples in the x, y directions determines the lowest spatial frequency we can resolve, $u_{min} = 1/(M\Delta x)$
- The sample interval, $\Delta x, \Delta y$, dictates the highest spatial frequency we can resolve, $u_{max} = 1/(2\Delta x)$
- The number of samples, M, N, dictates the number of spatial frequency 'bins' that can be resolved.

- *Addition & linearity* : as with continuous functions
- *Shift theorem* : : if $f(x) \to F(u)$ then $f(x-a) \to \exp[-i2\pi ua/M]F(u)$, hence

$$f(x,y)\exp[i(u_o x + v_o y)/M] \to F(u-u_o, v-v_o)$$

  and

$$f(x-x_o, y-y_o) \to F(u,v)\exp[-i2\pi(ux_o + vy_o)/M]$$

  if we let $u_o = v_o = M/2$ then we can shift the frequency space to the centre of the frequency square

$$f(x,y)(-1)^{x+y} \to F(u-M/2, v-M/2)$$

- *Discrete convolution* :

$$f(x) * g(x) = \sum_{m=0}^{M-1} f(m)g(x-m)$$

  and

$$DFT[f(x) * g(x)] = MF(u)G(u)$$

- *Power theorem* :

$$\sum_{x=0}^{M-1} |f(x)|^2 = M \sum_{u=0}^{M-1} |F(u)|^2$$

- *Periodicity* :

$$F(u,v) = F(u+M,v) = F(u,v+M) = F(u+M,v+M)$$

  Note that this leads us to deduce the *aliasing theorem*

- *Rotation* :

$$f(r, \theta+\theta_o) \to F(\omega, \phi+\theta_o)$$

- *Average value* : If we define the average value of the 2-D function as

$$\overline{f}(x,y) = \frac{1}{M^2} \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} f(x,y)$$

  then

$$\overline{f}(x,y) = \frac{1}{M} F(0,0)$$

- *Laplacian* : The Laplacian of a 2-D variable is defined as

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

  The DFT of the above is hence

$$-(2\pi)^2(u^2+v^2)F(u,v)$$

Trick : Plots of $|F(u,v)|$ often decay very rapidly from a central peak, so it is good to display on a log scale. Often the transform, $F'(u,v) = \log[1+|F(u,v)|]$ is used.

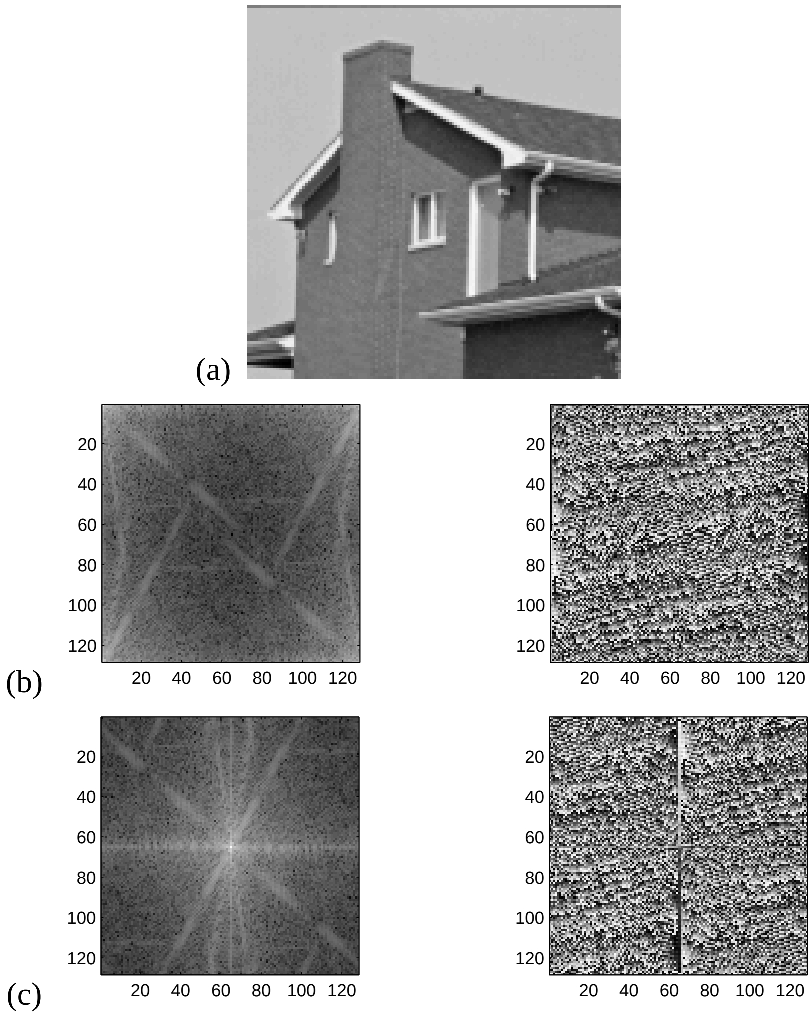


Figure 2.1: *Image (a), Fourier spectrum (b) and shifted Fourier spectrum (c).*

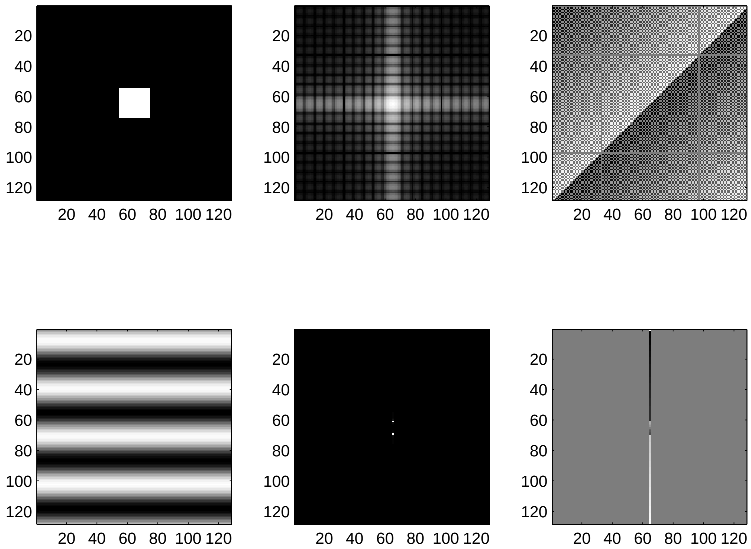


Figure 2.2: *Some 2-D functions and their resultant DFTs*

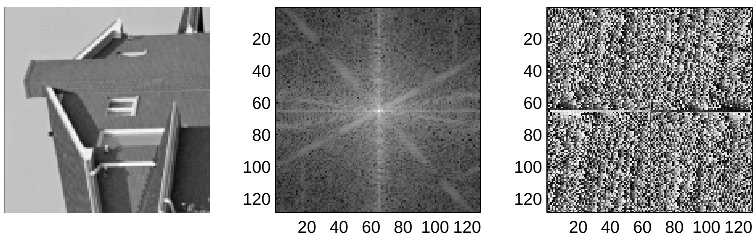


Figure 2.3: *Rotation in FT space.*

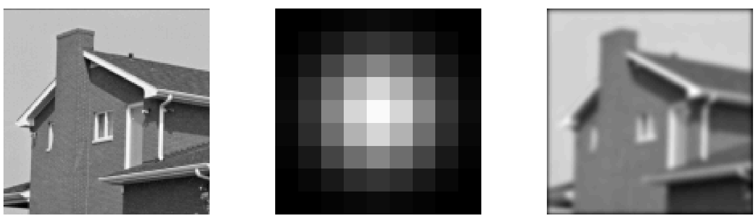


Figure 2.4: *2-D Convolution.*

2.2 Other transforms

The FT represents a specific case in a more general transform theory. In the FT the image is decomposed into a series of harmonic functions (sines and cosines). These have the property of being *orthogonal* functions and form a *complete basis* set. They are not the only such functions, however.

- FT kernel basis
- Walsh
- Hadamard
- Discrete cosine transform (DCT)
- Wavelets - localised functions.

We concentrate later in the course on the use of the DCT as this is the most widely used of the above and forms the basis of JPEG compression. We also look at wavelets as these form the basis of the JPEG-2000 compression scheme.

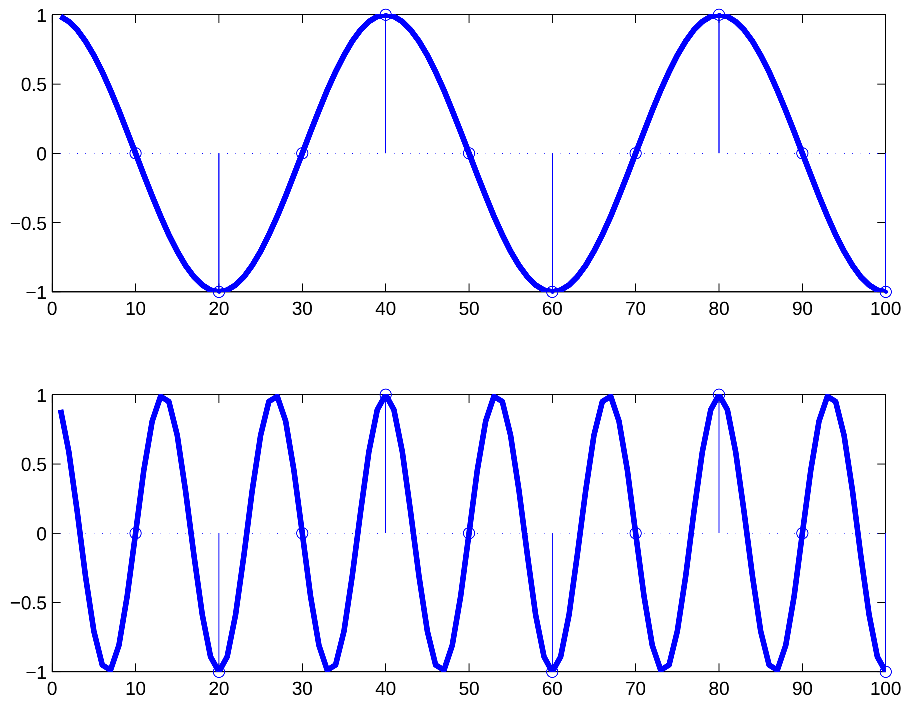
2.3 Digital filtering

2.3.1 The sampling process

This is performed by an *analogue to digital converter* (ADC) in which the continuous function $f(x)$ is replaced by a "discrete function" $f[k]$, which is defined only at $x = kT$, with $k = 0, 1, 2$. We thence only need consider the digitised sample set $f[k]$ and the sample interval T. A simple generalisation allows for a sampled set over the 2-D plane, with samples at $u\Delta M, v\Delta N$ so that u, v indexes the image pixels.

Aliasing

Consider $f(x) = \cos(\frac{\pi}{2}\frac{t}{T})$ (one cycle every 4 samples) and also $f(t) = \cos(\frac{3\pi}{2}\frac{x}{T})$ (3 cycles every 4 samples) as shown in the Figure. Note that the resultant samples are the same. This result is referred to as aliasing.



2.4 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 set of numbers produce a “filtered” version of that set? Consider the noisy signal of figure 2.5, together with its sampled version:

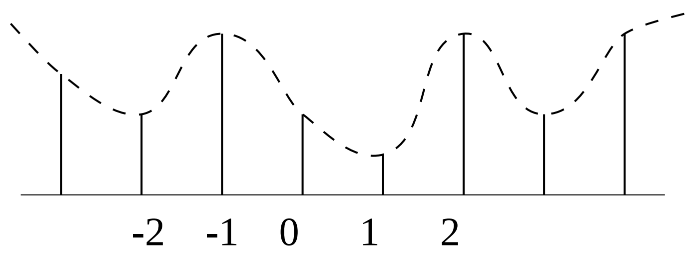


Figure 2.5: *Noisy data.*

One way to (e.g) 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. 2.6

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

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.

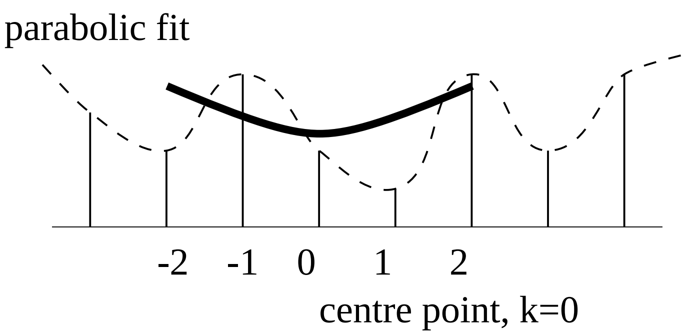


Figure 2.6: *Parabolic fit.*

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:

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

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

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

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. 2.7 shows this filtering effect. The magnitude response (which we will re-consider later) for the

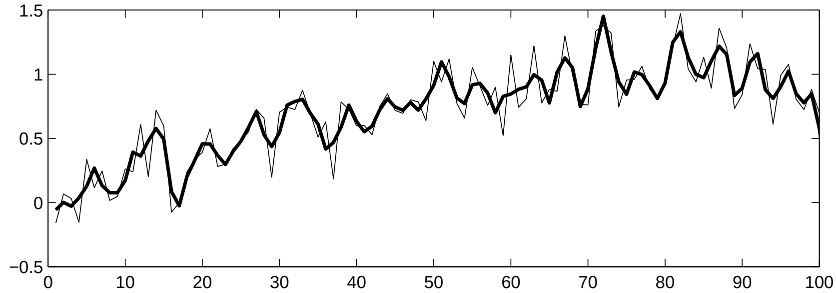


Figure 2.7: *Noisy data (thin line) and 5-point parabolic filtered (thick line).*

5-point parabolic filter is shown below in Fig. 2.8.

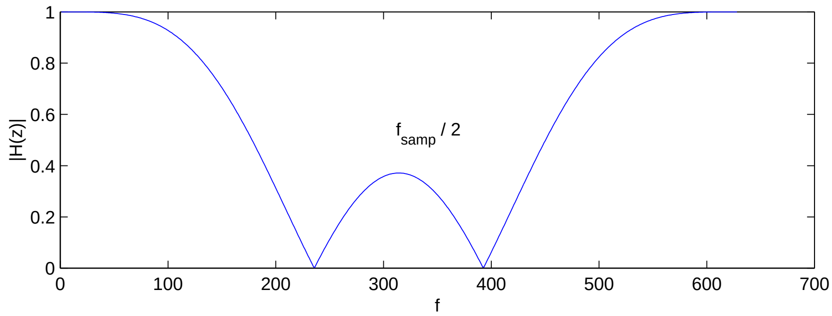


Figure 2.8: *Frequency response of 5-point parabolic filter.*

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 equation*):

$$r[k] = \sum_{i=0}^{N} a_i f[k-i]$$

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

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

This is a *non-causal* filter since a given output value $r[k]$ depends not only on previous inputs, but also on the current input $f[k]$, the input $f[k+1]$ and the input $f[k+2]$. The problem is solved by delaying the calculation of the output value $f[k]$ (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:

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

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

Proof:

Let $f[k] = 0$, except at $k = 0$, where $f[0] = 1$. Then $r[k] = \sum_i a_i f[k-i] = a_k x[0]$ (all terms zero except when $i = k$). This is equal to a_k since $f[0] = 1$. Therefore $r[0] = a_0$; $r[1] = a_1$; 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 $r[k]$ is also a function of previous outputs:

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

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

2.5 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 $x = kT$ is e^{-kTs}. Consider the discrete function $f[k]$ to be a succession of impulses, for example of area $f(0)$ occurring at $x = 0$, $f(1)$ occurring at $x = T$, 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}.

2.5.1 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 $x = T$, $\ldots a_i$ at $x = nT$ with $n = 0$ 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:

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

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

$$G(z)F(z) = (a_0 + a_1 z^{-1} + \ldots a_n z^{-n} + \ldots a_N z^{-N})(f[0] + f[1]z^{-1} + \ldots f[k]z^{-k})$$

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

$$a_0 f[k] + a_1 f[k-1] + \ldots a_n f[k-n] + \ldots a_N f[k-N]$$

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

For non-recursive filters:

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

For recursive filters:

$$R(z) = \sum_{n=0}^{N} a_i z^{-i} F(z) + \sum_{i=m}^{M} b_i z^{-i} R(z)$$

$$G(z) = \frac{R(z)}{F(z)} = \frac{\sum_n a_n z^{-n}}{1 - \sum_m b_m z^{-m}}$$

2.5.2 z-plane pole-zero plot

Let $z = e^{sT}$, where T = sampling period. Since $s = \sigma + i2\pi u$, we have:

$$z = e^{\sigma T} e^{i2\pi uT}$$

If $\sigma = 0$, then $\mid z \mid = 1$ and $z = e^{i2\pi uT} = \cos 2\pi uT + i \sin 2\pi uT$, 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.

2.6 Frequency response of a digital filter

This can be obtained by evaluating the (pulse) transfer function on the unit circle (i.e. $z = e^{2\pi iuT}$).

Proof
Consider the general filter

$$r[k] = \sum_{n=0}^{\infty} a_n f[k-i]$$

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

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

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

$$\text{ie.} r[k] = \frac{1}{2}\sum_{n=0}^{\infty} a_n e^{i\{2\pi u[k-n]T+\theta\}} + \frac{1}{2}\sum_{n=0}^{\infty} a_n e^{-i2\pi\{u[k-n]T+\theta\}}$$

$$= \frac{1}{2}e^{i(2\pi ukT+\theta)}\sum_{n=0}^{\infty} a_n e^{-i2\pi unT} + \frac{1}{2}e^{-i2\pi(ukT+\theta)}\sum_{n=0}^{\infty} a_n e^{i2\pi unT}$$

$$\text{Now} \sum_{n=0}^{\infty} a_n e^{-i2\pi unT} = \sum_{n=0}^{\infty} a_n (e^{i2\pi uT})^{-n}$$

$$\text{But } G(z) \text{ for this filter is} \qquad \sum_{n=0}^{\infty} a_n z^{-n}$$

$$\text{and so} \qquad \sum_{n=0}^{\infty} a_n e^{-i2\pi unT} = G(z)_z = e^{i2\pi uT}$$

Let $G(z)_z = e^{i2\pi uT} = Ae^{i\phi}$.
Then

$$\sum_{n=0}^{\infty} a_n e^{i2\pi unT} = Ae^{-i\phi} \qquad \text{(complex conjugate)}$$

Hence $r[k] = \frac{1}{2}e^{i(2\pi ukT+\theta)}Ae^{i\phi} + \frac{1}{2}e^{-i(2\pi ukT+\theta)}Ae^{-i\phi}$

$$\text{or} \quad r[k] = A\cos(2\pi ukT + \theta + \phi) \qquad \text{when } f[k] = \cos(2\pi ukT + \theta)$$

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

$$G(z)|_{z=e^{i2\pi uT}}$$