Review Notes: Digital Signal Processing (EE3-07)

Aidan O. T. Hogg {aidan.hogg13@imperial.ac.uk}

Imperial College London, (last updated: November 22, 2018)

*This document is in **no way complete**, it was created merely to aid me in the running of the tutorials for 'EE3-07 - Digital Signal Processing'. I have only made it available in the hope others find it useful. All of the content, including every single figure, was created by me so PLEASE email me if you find any mistakes! Furthermore, the whole of this document was highly influenced by the references which are listed at the end.*

Contents

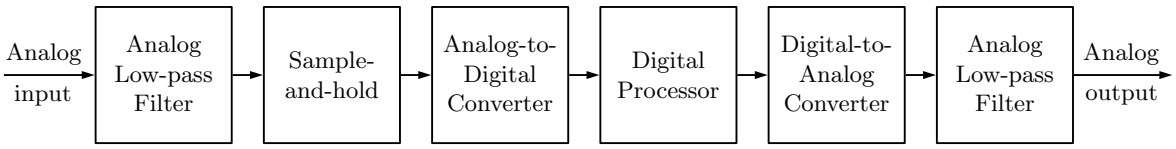
1 Module 1 - Sampling, Z-Transforms and System Functions

1.1 Types of Signals

There are a small number of special sequences that are worth taking note of:

- Unit impulse: $\delta[n] = u[n] - u[n-1] = \begin{cases} 1, & n = 0 \\ 0, & \text{otherwise} \end{cases}$
- Unit step: $u[n] = \sum_{k=0}^{\infty} \delta[n-k] = \begin{cases} 1, & n \geq 0 \\ 0, & \text{otherwise} \end{cases}$
- Right-sided: $x[n] = 0$ for $n < N_{\min}$
- Left-sided: $x[n] = 0$ for $n > N_{\max}$
- Finite length: $x[n] = 0$ for $n \notin [N_{\min}, N_{\max}]$
- Causal: $x[n] = 0$ for $n < 0$
- Anticausal: $x[n] = 0$ for $n > 0$
- Finite Energy: $\sum_{n=-\infty}^{\infty} |x[n]|^2 < \infty$
- Absolutely Summable: $\sum_{n=-\infty}^{\infty} |x[n]| < \infty \implies$ Finite Energy

1.2 Effect of Sampling



It is now common practice to process analog signals digitally due to the many advantages of digital signal processing. This process is shown above. First an analog low-pass filter is used to remove any frequencies that could cause aliasing and thus is aptly named the 'anti-aliasing filter'. Secondly the anolog signal is sampled and held to allow time for the analog-to-digital conversion to take place. The output analog-to-digital converter is then a digital signal that can be processed using different digital signal processing techniques including: delaying, multiplying and adding to extract the desired information from the signal. This desired signal can then be converted back into an analog signal using a digital-to-analog converter. The ouput of the digital-to-analog converter is then passed through a low-pass filter to remove higher frequency components that are created by the sampling process; this filter is normally referred to as the reconstruction filter due to the fact that it reconstructs the analog signal at the output.

Let $x_a(t)$ be a analog signal that is sampled at internals of T, creating a signal $x_s(t)$ where:

$$x_s(t) = x_a(nT), \quad -\infty < n < \infty, \quad T > 0$$

with the reciprocal of T being the *sampling frequency* F_s.

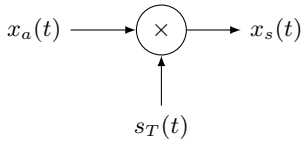
Now let $x_a(t)$ have the Fourier transform $X_a(j\Omega)$, where Ω denotes the analog radian frequency ($\Omega = 2\pi F$), defined by:

$$X_a(j\Omega) = \int_{-\infty}^{\infty} x_a(t) e^{-j\Omega t} dt$$

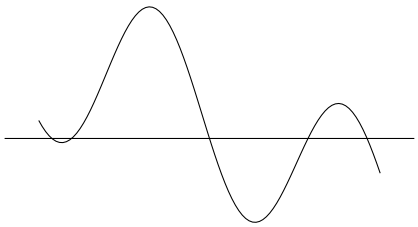
So in order to discover the interrelationship between $X_a(j\Omega)$ and $X_s(j\Omega)$, we need to idealize the sampling waveform to a sequence of impulses, given by $s_T(t)$:

$$s_T(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$$

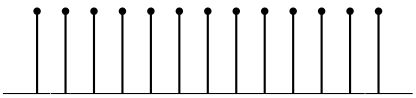
Therefore $x_s(t) = x_a(t)s_T(t)$ which is illustrated in the figure below.



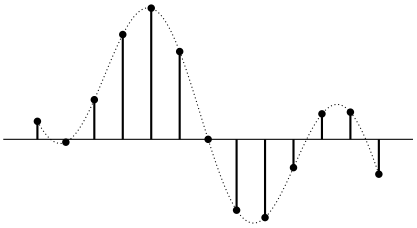
To get a better conceptual understanding of this sampling process it has been visualised below.



(a) $x_a(t)$: continuous-time signal



(b) $s_T(t)$: impluse train



(c) $x_s(t)$: sampled signal

Since $S_T(t)$ is a periodic waveform it can be written as a Fourier series:

$$s_T(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) = \frac{1}{T} \sum_{k=-\infty}^{\infty} e^{jk\Omega_s t}$$

This result comes from the compact representation of the Fourier Series which uses complex exponentials:

$$s_T(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\Omega_s t}, \quad \text{where } c_k = \frac{1}{T}\int_{-T/2}^{T/2} s_T(t)e^{-jk\Omega_s t}dt = \frac{1}{T}\int_{-T/2}^{T/2} \delta(t)e^{-jk\Omega_s t}dt = \frac{1}{T}$$

This means that $x_s(t)$ can be rewritten in the following way:

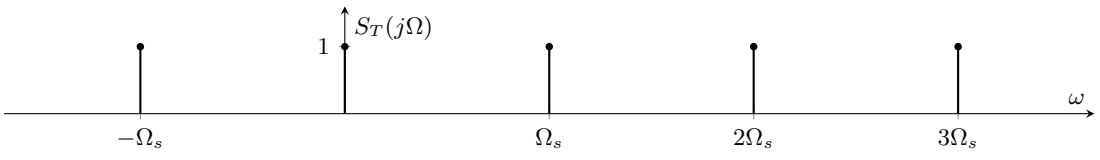
$$x_s(t) = x_a(t)s_T(t) = \sum_{n=-\infty}^{\infty} x_a(t)\delta(t - nT) = x_a(t)\frac{1}{T}\sum_{k=-\infty}^{\infty} e^{jk\Omega_s t} = \frac{1}{T}\sum_{k=-\infty}^{\infty} x_a(t)e^{jk\Omega_s t}$$

By taking the Fourier Transform of both sides it is shown that:

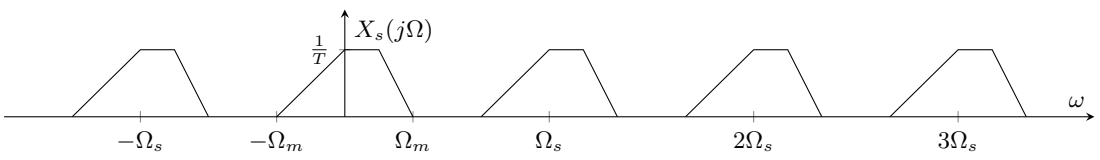
$$X_s(j\Omega) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_a(j(\Omega + k\Omega_s))$$

Thus it can be seen that $X_s(j\Omega)$ is a periodic function of frequency Ω, consisting of the sum of shifted and scaled replicas of $X_a(j\Omega)$, shifted by integer multiples of Ω_s and scaled by $\frac{1}{T}$.

(a) Spectrum of the original continuous-time signal $x_a(t)$



(b) Spectrum of the sequence of impulses $s_T(t)$



(c) Spectrum of the sampled signal $x_s(t)$

1.3 Time Scaling

It is normal to scale the time so that $F_s = 1$ which results in dividing all real frequencies and angular frequencies by F_s and dividing all real times by T.

For example, instead of designing a 1 kHz low-pass filter for $f_s = 44.1$ kHz it would often be easier to design a 0.0227 kHz filer for $f_s = 1$ kHz.

It is common to use F for 'real' frequencies and Ω for 'real' angular frequencies. The scaled versions being f for normalised frequency and ω for normalised angular frequency where the units of ω are 'radians per sample' and f are 'samples per second'. Therefore:

$$\omega = \frac{\Omega}{F_s} = \frac{2\pi\Omega}{\Omega_s} = \frac{2\pi F}{F_s} = 2\pi f, \quad \text{where } f = \frac{F}{F_s}$$

thus

$$X_s(j\Omega) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_a(j(\Omega + k\Omega_s))$$

can be written as

$$X\left(e^{j\omega}\right) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_a\left(j\left(\frac{\omega}{T} + \frac{2\pi k}{T}\right)\right) \quad \text{where } \Omega = \frac{\omega}{T} \text{ and } \Omega_s = 2\pi F_s = \frac{2\pi}{T}$$

Warning: several MATLAB routines scale time so that fs = 2 Hz. Non-standard, weird and irritating.

1.4 Sampling Theorem

Sampling has the effect of creating spectral images every Ω_s and, therefore, to avoid information loss due to overlapping images (aliasing) the following condition must be met:

Definition:

$$|\Omega_m| \leq \frac{\Omega_s}{2}, \quad \text{where } \Omega_s = 2\pi F_s = 2\pi/T \quad \Longrightarrow \quad |\omega| \leq \pi$$

The frequency $2\Omega_m$ is called the Nyquist rate. Sampling above this frequency is called *oversampling*, conversely, sampling below this frequency is called *undersampling*. Lastly sampling at a frequency exactly equal to the Nyquist rate is called *critical sampling*.

1.5 Recovery of the Analog Signal

It can be seen earlier that the continuous-time signal $x_a(t)$ can be recovered by passing the signal $x_s(t)$ through an ideal low-pass filter $H(j\Omega)$ with cuffoff frequency Ω_c, where $\Omega_m < \Omega_c < (\Omega_s - \Omega_m)$ to avoid aliasing.

The impulse response of h(t) of this low-pass filter can be obtained by taking the inverse Fourier transform of its frequency response $H(j\Omega)$:

$$H(j\Omega) = \begin{cases} T, & |\Omega| \leq \Omega_c \\ 0, & |\Omega| > \Omega_c \end{cases}$$

which is given by

$$h(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} H(j\Omega)e^{j\Omega t}d\Omega = \frac{T}{2\pi}\int_{-\Omega_c}^{\Omega_c} e^{j\Omega t}d\Omega = \frac{T}{2\pi}\Big[\frac{1}{jt}e^{j\Omega t}\Big]_{-\Omega_c}^{\Omega_c} = \frac{T}{2\pi}\Big(\Big[\frac{1}{jt}e^{j\Omega_c t}\Big] - \Big[\frac{1}{jt}e^{-j\Omega_c t}\Big]\Big)$$

$$= \frac{T}{j2\pi t}\Big(e^{j\Omega_c t} - e^{-j\Omega_c t}\Big) = \frac{T}{\pi t}\sin(\Omega_c t) = \frac{\sin(\Omega_c t)}{\Omega_s t/2}, \quad -\infty < t < \infty$$

see also that $x_s(t)$ is given by

$$x_s(t) = \sum_{n=-\infty}^{\infty} x[n]\delta(t - nT), \quad \text{where } x[n] = x_a(nT)$$

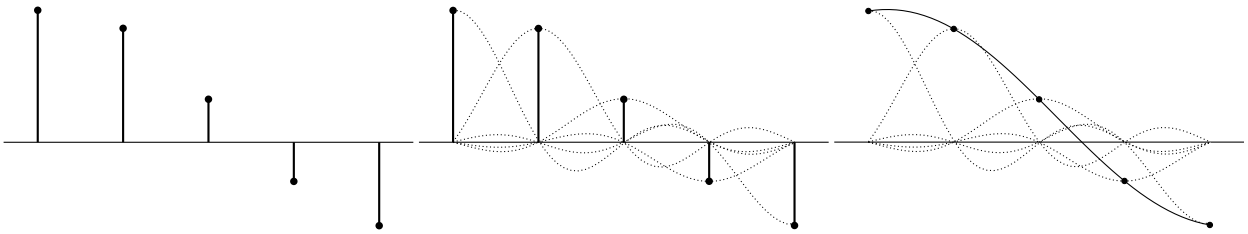
Therefore the output of the low-pass filter $\hat{x}_a(t)$ is given by $x_s(t) * h(t)$, which can be written as

$$\hat{x}_a(t) = \sum_{n=-\infty}^{\infty} x[n]h(t - nT)$$

Assuming for simplicity that $\Omega_c = \Omega_s/2 = (2\pi F_s)/2 = \pi/T$ by substitution of h(t) we arrive at

$$\hat{x}_a(t) = \sum_{n=-\infty}^{\infty} x[n]\frac{\sin[\pi(t - nT)/T]}{\pi(t - nT)/T} = \sum_{n=-\infty}^{\infty} x[n]\,\mathrm{sinc}\left[\frac{(t - nT)}{T}\right]$$

as a result, $\hat{x}_a(nT) = x[n] = x_a(nT)$ for $n \in \mathbb{Z}$ in the range $-\infty < n < \infty$, whether or not the Nyquist theorem has been satisfied, however, $\hat{x}_a(t) = x_a(t)$ is only true if $\Omega_s \geq 2\Omega_m$ (Nyquist condition).



(a) $x[n]$: sampled signal (b) sinc interpolation (c) $\hat{x}_a(t)$: reconstructed signal

1.6 Z-Transform

The z-transform converts a discrete-time signal, $x[n]$, into a function, $X(z)$, of an arbitrary complex-valued variable z.

Definition:

$$\mathcal{Z}\{x[n]\} = X(z) = \sum_{n=-\infty}^{\infty} x[x]z^{-1} \tag{1}$$

which only exists for particular values of z for which the series converges

The values of z for which $X(z)$ converges is called the 'Region of Convergence (ROC)'. $X(z)$ will always converge absolutely inside the ROC and may converge on some, all, or none of the boundary.

- Converge absolutely $\Leftrightarrow \sum_{n=-\infty}^{\infty} |x[n]z^n| < \infty$
- Absolutely summable $\Leftrightarrow X(z)$ converges for $|z| = 1$
- Causal $\Rightarrow X(\infty)$ converges
- Anticausal $\Rightarrow X(0)$ converges

Geometric series:

$$1 + r + r^2 + r^3 + \cdots = \frac{1}{1-r}, \quad \text{for } |r| < 1$$

Proof:

$$\begin{aligned} s &= 1 + r + r^2 + r^3 + \cdots \\ rs &= r + r^2 + r^3 + r^4 + \cdots \\ s - rs &= 1, \quad \text{given } |r| < 1 \\ s &= \frac{1}{1-r}, \quad \text{for } |r| < 1 \end{aligned}$$

Examples:

Z-transform of a causal sequence

Consider the causal signal $x[n] = \alpha^n u[n]$ then

$$X(z) = \mathcal{Z}\{x[n]\} = \sum_{n=-\infty}^{\infty} x[n] z^{-n} = \sum_{n=-\infty}^{\infty} \alpha^n u[n] z^{-n} = \sum_{n=0}^{\infty} \alpha^n z^{-n}$$

The above power series converges to

$$X(z) = \frac{1}{1-\alpha z^{-1}}, \quad \text{for } |\alpha z^{-1}| < 1 \quad \Rightarrow \quad |z| > |\alpha|$$

Z-transform of an anticausal sequence

Consider the anticausal signal $x[n] = -\alpha^n u[-n-1]$

$$X(z) = \mathcal{Z}\{x[n]\} = \sum_{n=-\infty}^{\infty} x[n] z^{-n} = \sum_{n=-\infty}^{\infty} -\alpha^n u[-n-1] z^{-n} = -\sum_{n=-\infty}^{-1} \alpha^n z^{-n} = -\sum_{m=1}^{\infty} \alpha^{-m} z^m$$

The above power series converges to

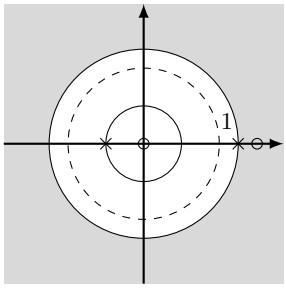
$$X(z) = -\frac{\alpha^{-1} z}{1-\alpha^{-1} z} = \frac{1}{1-\alpha z^{-1}}, \quad \text{for } |\alpha^{-1} z| < 1 \quad \Rightarrow \quad |z| < |\alpha|$$

Z-transform of a rational polynomial

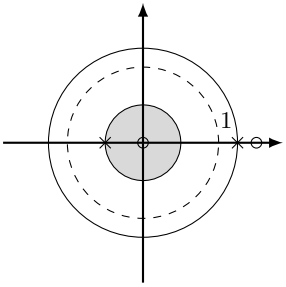
$$H(z) = \frac{8 - 12z^{-1}}{8 - 6z^{-1} - 5z^{-2}} = \frac{8z^2 - 12z}{8z^2 - 6z - 5} = \frac{z(z-1.5)}{(z+0.5)(z-1.25)} \Rightarrow \begin{array}{l} \text{Zeros at } z = \{0, +1.5\} \\ \text{Poles at } z = \{-0.5, +1.25\} \end{array}$$

$$\text{Partial Fractions: } H(z) = \frac{16/7}{(2+z^{-1})} - \frac{4/7}{(4-5z^{-1})} = \frac{8}{7} \times \frac{1}{(1+\frac{1}{2}z^{-1})} - \frac{1}{7} \times \frac{1}{(1-\frac{5}{4}z^{-1})}$$

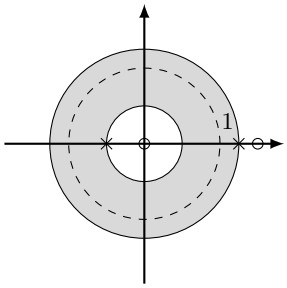
Warning: The zeros at $z = 0$ can be easily missed when $H(z)$ is written as a function of z^{-1}.



(a) Right-sided and unstable



(b) Left-sided and unstable



(c) Two-sided and stable

(a) For the ROC defined by $|z| > |1.25|$, the impulse response is a right-sided sequence given by the sum of two causal sequences which is given by:

$$h[n] = \frac{8}{7}\Big(-\frac{1}{2}\Big)^n u[n] - \frac{1}{7}\Big(\frac{5}{4}\Big)^n u[n]$$

Since the second sequence on the right-hand side in the above equation is not absolutely summable the LTI system is unstable.

(b) For the ROC defined by $|z| < |0.5|$, the impulse response is a left-sided sequence given by the sum of two anticausal sequences which is given by:

$$h[n] = \frac{8}{7}\Big(\frac{1}{2}\Big)^n u[-n-1] - \frac{1}{7}\Big(-\frac{5}{4}\Big)^n u[-n-1]$$

Since the first sequence on the right-hand side in the above equation is not absolutely summable the LTI system is unstable.

(c) For the ROC defined by $|0.5| < |z| < |1.25|$, the impulse response is a two-sided sequence given by the sum of a causal sequence and an anticausal sequence which is given by:

$$h[n] = \frac{8}{7}\Big(-\frac{1}{2}\Big)^n u[n] - \frac{1}{7}\Big(-\frac{5}{4}\Big)^n u[-n-1]$$

Both of the sequences in the above equation are absolutely summable so, therefore, the LTI system is stable. Note that the ROC contains the unit circle.

1.7 Inverse Z-Transform

Definition:

$$x[n] = \frac{1}{2\pi j}\oint X(z)z^{n-1}dz \tag{2}$$

where the integral is anti-clockwise around a circle within the ROC, $z = Re^{j\theta}$

Proof:

$$\begin{aligned}\frac{1}{2\pi j}\oint X(z)z^{n-1}dz &= \frac{1}{2\pi j}\oint\Big(\sum_{m=-\infty}^{\infty} x[m]z^{-m}\Big)z^{n-1}dz \\ &\overset{\text{(i)}}{=} \sum_{m=-\infty}^{\infty} x[m]\frac{1}{2\pi j}\oint z^{n-m-1}dz \\ &\overset{\text{(ii)}}{=} \sum_{m=-\infty}^{\infty} x[m]\delta[n-m] = x[n]\end{aligned}$$

(i) Depends on the circle with radius R lying within the ROC

(ii) Cauchy's theorem: $\frac{1}{2\pi j}\oint z^{k-1}dz = \delta[k]$ for $z = Re^{j\theta}$ anti-clockwise

$$\begin{aligned}\frac{dz}{d\theta} = jRe^{j\theta} \Rightarrow \frac{1}{2\pi j}\oint z^{k-1}dz &= \frac{1}{2\pi j}\int_{\theta=0}^{2\pi} R^{k-1}e^{j(k-1)\theta}\times jRe^{j\theta}d\theta \\ &= \frac{R^k}{2\pi}\int_{\theta=0}^{2\pi} e^{jk\theta}d\theta \\ &= R^k\delta(k) = \delta(k) \quad [R^0 = 1]\end{aligned}$$

In practice the inverse z-transform is found using a combination of partial fractions and a table of z-transforms.

1.8 Energy and Power for Discrete-Time Signals

The **energy** of discrete-time signal , $x[n]$, is given by

$$E = \sum_{n=-\infty}^{\infty} |x[n]|^2 \text{ and if the signal is finite equals } E_N = \sum_{n=-N}^{N} |x[n]|^2$$

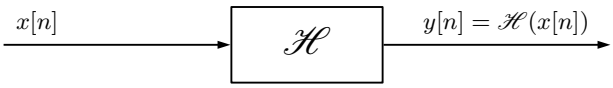
The **power** of discrete-time signal , $x[n]$, is the average of $x^2[n]$ in 'energy per sample'

$$P = \lim_{N\to\infty} \frac{1}{(2N+1)} \sum_{n=-N}^{N} |x[n]|^2 = \lim_{N\to\infty} \frac{1}{(2N+1)} E_N$$

Note: P is same value as the power of pre-sampled signal $x(t)$ in 'energy per second' provided there is no aliasing.

A discrete-time energy signal is defined as one for which $0 < E < \infty$ and a discrete-time power signal is defined as one for which $0 < P < \infty$. It is possible for a discrete-time signal to be neither an energy signal nor a power signal.

1.9 Linear Time-Invariant Systems



A linear time-invariant system can be defined by two properties:

Linear: $\mathscr{H}(\alpha u[n] + \beta v[n]) = \alpha\mathscr{H}(u[n]) + \beta\mathscr{H}(v[n])$

Time Invariant: $y[n] = \mathscr{H}(x[n]) \Rightarrow y[n-r] = \mathscr{H}(x[n-r]) \forall r$

Note: The behaviour of an LTI system is completely defined by its impulse response: $h[n] = \mathscr{H}(\delta[n])$

Proof:

$$x[n] = \sum_{r=-\infty}^{\infty} x[r]\delta[n-r]$$

$$\mathscr{H}\Big(x[n]\Big) = \mathscr{H}\Big(\sum_{r=-\infty}^{\infty} x[r]\delta[n-r]\Big) = \sum_{r=-\infty}^{\infty} x[r]\mathscr{H}(\delta[n-r])$$

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

$$= x[n] * h[n]$$

1.10 BIBO Stability Condition

BIBO Stability: Bounded Input, $x[n] \Rightarrow$ Bounded Output, y[n].

These statements are equivalent:

(a) An LTI system is BIBO stable
(b) The impulse response, h[n], is absolutely summable, that is to say, $\sum_{n=-\infty}^{\infty} |h[n]| < \infty$
(c) The region of absolute convergence of the transfer function, $H(z)$, includes the unit circle

Proof (a) $\Rightarrow$ (b)

Suppose the output $y[n]$ corresponding to the input $x[n]$ is given by the convolution sum:

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

Suppose that the input is bounded with bound M, then:

$$|y[n]| \leq \sum_{r=-\infty}^{\infty} |x[n-r]||h[r]| \leq M \sum_{r=-\infty}^{\infty} |h[r]|$$

If (a) is true then the output is bounded with a bound N:

$$|y[n]| \leq M \sum_{r=-\infty}^{\infty} |h[r]| = N < \infty$$

Thus the impulse response, $h[r]$, has to be absolutely summable.

Proof (b) $\Rightarrow$ (a)

Let's again suppose the output $y[n]$ is given by the convolution sum:

$$|y[n]| = \Big| \sum_{r=-\infty}^{\infty} x[n-r]h[r] \Big| \leq \sum_{r=-\infty}^{\infty} |x[n-r]||h[r]|$$

Suppose the impulse response is bounded by S and the input is bounded by M:

$$|y[n]| \leq M \sum_{r=-\infty}^{\infty} |h[r]| = MS < \infty$$

2 Module 2 - Discrete Fourier Transform

2.1 Different Types of Fourier Transforms

Definitions:

- CTFT (Continuous-Time Fourier Transform): $x(t) \rightarrow X(j\Omega)$

  Forward Transform: $X(j\Omega) = \int_{-\infty}^{\infty} x(t)e^{-j\Omega t}dt$, Inverse Transform: $x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\Omega)e^{j\Omega t}d\Omega$

- DTFT (Discrete-Time Fourier Transform): $x[n] \rightarrow X(e^{j\omega})$

  Forward Transform: $X(e^{j\omega}) = \sum_{-\infty}^{\infty} x[n]e^{-j\omega n}$, Inverse Transform: $x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n}d\omega$

- DFT (Discrete Fourier Transform): $x[n] \rightarrow X[k]$

  Forward Transform: $X[k] = \sum_{0}^{N-1} x[n]e^{-j\frac{2\pi k}{N}n}$, Inverse Transform: $x[n] = \frac{1}{N}\sum_{0}^{N-1} X[k]e^{j\frac{2\pi k}{N}n}$

2.2 Convergence of the DTFT

The DTFT, $X(e^{j\omega}) = \sum_{-\infty}^{\infty} x[n]e^{-j\omega n}$, does not converge for all $x[n]$.

Consider the finite sum: $X_N(e^{j\omega}) = \sum_{-N}^{N} x[n]e^{-j\omega n}$

Strong convergence:

$x[n]$ absolutely summable $\Rightarrow X(e^{j\omega})$ converges uniformly:

$$\sum_{-\infty}^{\infty} |x[n]| < \infty \Rightarrow \lim_{N\to\infty} \left\{\sup_{\omega}|X(e^{j\omega}) - X_N(e^{j\omega})|\right\} = 0$$

Weaker convergence:

$x[n]$ finite energy $\Rightarrow X(e^{j\omega})$ converges in the mean square:

$$\sum_{-\infty}^{\infty} |x[n]|^2 < \infty \Rightarrow \lim_{N\to\infty} \int_{-\pi}^{\pi} |X(e^{j\omega}) - X_N(e^{j\omega})|^2 d\omega = 0$$

2.3 Properties of the DTFT

1. The DTFT is periodic in ω
$$X\left(e^{j(\omega+2m\pi)}\right) = X(e^{j\omega}) \;\forall\; m$$

2. The DTFT is just the z-transform evaluated on the unit circle, i.e. $z = e^{j\omega}$
$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$$

3. The DTFT is the same as the CTFT of a signal comprising of impulses at the sample times
$$x_s(t) = \sum_{n=-\infty}^{\infty} x[n]\delta(t-nT) = x_a(t) \times \sum_{n=-\infty}^{\infty} \delta(t-nT)$$

Proof:

$$\begin{aligned} X(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} = \sum_{n=-\infty}^{\infty} x[n] \int_{-\infty}^{\infty} \delta(t-nT)e^{-j\omega\frac{t}{T}}dt \\ &= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x[n]\delta(t-nT)e^{-j\omega\frac{t}{T}}dt, \quad \text{allowed if } \sum_{n=-\infty}^{\infty} |x[n]| < \infty \\ &= \int_{-\infty}^{\infty} x_s(t)e^{j\Omega t}dt, \quad \text{due to } \omega = \Omega T \end{aligned}$$

2.4 DFT & DTFT

It is helpful to note that the DFT, $X[k] = \sum_0^{N-1} x[n]e^{-j\frac{2\pi k}{N}n}$, is the same as DTFT in certain cases:

Case 1: $x[n] = 0$ for $n \notin [0, N-1]$ DFT is the same as DTFT at $\omega_k = \frac{2\pi}{N}k$

Case 2: $x[n]$ is periodic with period N DFT equals the normalised DTFT

2.5 Symmetries

It is useful to note that if a signal $x[n]$ has a special property in the time domain then there will be a corresponding property in the frequency domain, $X(e^{j\omega})$ and $X[k]$.

One Domain	**Other Domain**
Discrete	Periodic
Symmetric	Symmetric
Antisymmetric	Antisymmetric
Real	Conjugate Symmetric
Imaginary	Conjugate Antisymmetric
Real & Symmetric	Real & Symmetric
Real & Antisymmetric	Imaginary & Antisymmetric

Symmetric: $x[n] = x[-n]$

$X(e^{j\omega}) = X(e^{-j\omega})$

$X[k] = X[(-k)_{\text{mod}N}] = X[N-k]$ for $k > 0$

Conjugate Symmetric: $x[n] = x^*[-n]$

Conjugate Antisymmetric: $x[n] = -x^*[-n]$

2.6 Parseval's Relation

Parseval's relation states that all Fourier transforms preserve 'energy'.

- CTFT $\int_{-\infty}^{\infty} |x(t)|^2 dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} |X(j\Omega)|^2 d\Omega$
- DTFT $\sum_{-\infty}^{\infty} |x[n]|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |X(e^{j\omega})|^2 d\omega$
- DFT $\sum_0^{N-1} |x[n]|^2 = \frac{1}{N}\sum_0^{N-1} |X[k]|^2$

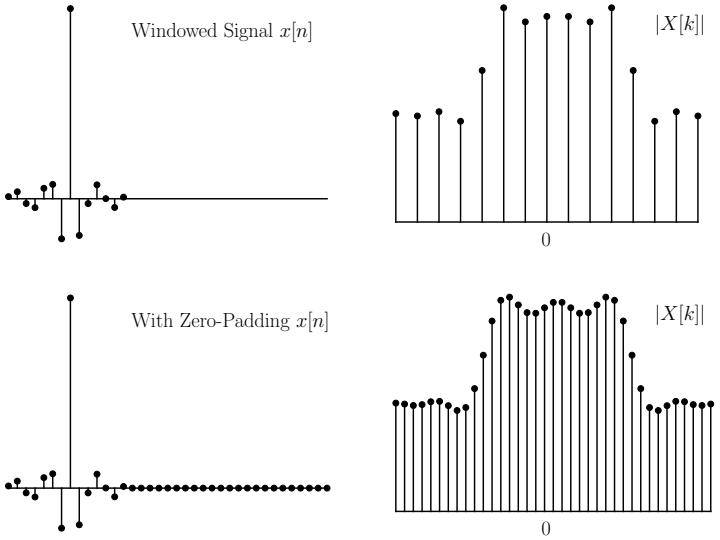
More generally, these transforms actually preserve complex inner products:

$$\sum_0^{N-1} x[n]y^*[n] = \frac{1}{N}\sum_0^{N-1} X[k]Y^*[k]$$

2.7 Zero-Padding

Zero padding is the process of added extra zeros onto the end of $x[n]$ before performing the DFT.

- Zero-padding causes the DFT to evaluate the DTFT at more values of ω_k. Denser frequency samples.
- Width of the peaks remains constant as they are determined by the length and shape of the window.
- Smoother graph but the increased frequency resolution is an **illusion**.



2.8 Matrix Interpretation of the FFT Algorithm

This is only for additional insight into how matrices play a very important role in DSP. It will also hopefully give you a better understanding of the FFT.

Here is the DFT matrix of F_N for $N = 4$. Where $w = e^{-j\frac{2\pi}{N}}$ (i.e $w = e^{-j\frac{2\pi}{4}} = -j$)

$$\text{DFT Matrix: } F_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & w & w^2 & w^3 \\ 1 & w^2 & w^4 & w^6 \\ 1 & w^3 & w^6 & w^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

The columns of F_4 are all orthogonal, this means the inner product of any column with any other column should be zero.

In the case of column 0 and 1 this is true: $(\text{column } 0)^T(\text{column } 1) = 1 - j - 1 + j = 0$

However, the inner product of column 1 and 3 apears to be 4: $(\text{column } 1)^T(\text{column } 3) = 1+1+1+1 = 4$

But this is wrong. These vectors are complex vectors, therefore, to get the correct inner product we must take the complex conjugate of one of the vectors: $(\overline{\text{column } 1})^T(\text{column } 3) = 1 + (j \cdot j) + 1 + (-j \cdot -j) = 0$

The inner product of every vector with itself is: $1 + 1 + 1 + 1 = 4$. All the vectors of F_4 have length $\sqrt{4} = 2$

Multiplying F_4 times $\frac{1}{4}\overline{F}_4$ produces I:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & (-j) & (-j)^2 & (-j)^3 \\ 1 & (-j)^2 & (-j)^4 & (-j)^6 \\ 1 & (-j)^3 & (-j)^6 & (-j)^9 \end{bmatrix} \times \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & j^2 & j^3 \\ 1 & j^2 & j^4 & j^6 \\ 1 & j^3 & j^6 & j^9 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Thus F_4^{-1} is $\frac{1}{4}\overline{F}_4^T$ also written as $\frac{1}{4}F_4^*$

General rule for F_N:

The columns of $\frac{1}{\sqrt{N}}F_N$ are orthogonal. Their inner products produce I.

$(\frac{1}{\sqrt{N}}\overline{F}_N^T)(\frac{1}{\sqrt{N}}F_N) = I$ means that the inverse is $F_N^{-1} = \frac{1}{N}\overline{F}_N^T = \frac{1}{N}F_N^*$ (Note: the DFT matrix is symmetric, so transposing has no effect. The inverse matrix just divides by N and replaces j by $-j$)

Fast Fourier Transform (FFT)

To reconstruct $X(k)$ we want to multiply by F_N times x as quickly as possible. The matrix has N^2 entries so we would normally have to perform N^2 separate multiplications. However, we can do better! The key to the idea is to connect F_N with the half-size DFT matrix $F_{N/2}$.

Assume N is a power of 2 (say $N = 2^{10} = 1024$). we can connect F_{1024} to F_{512} or rather two copies of F_{512}. When $N = 4$:

$$F_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & (-j) & (-j)^2 & (-j)^3 \\ 1 & (-j)^2 & (-j)^4 & (-j)^6 \\ 1 & (-j)^3 & (-j)^6 & (-j)^9 \end{bmatrix} \text{ and } \begin{bmatrix} F_2 & 0 \\ 0 & F_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & & \\ 1 & -1 & & \\ & & 1 & 1 \\ & & 1 & -1 \end{bmatrix}$$

Key idea:

$$F_4 = \begin{bmatrix} 1 & & 1 & \\ & 1 & & -j \\ 1 & & -1 & \\ & 1 & & j \end{bmatrix} \begin{bmatrix} 1 & 1 & & \\ 1 & -1 & & \\ & & 1 & 1 \\ & & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & & 1 & \\ & 1 & & \\ & & & 1 \end{bmatrix}$$

The permutation matrix on the right puts x_0 and x_2 (evens) ahead of x_1 and x_3 (odds). The matrix in the middle performs separate half-size transforms on the even and odds. The matrix on the left combines the two half-size outputs to get the correct full-size out $X[k] = F_4x[n]$.

You should check the result yourself by multiplying the three matrices together to get F_4.

The same idea applies when $N = 1024$:

$$F_{1024} = \begin{bmatrix} I_{512} & D_{512} \\ I_{512} & -D_{512} \end{bmatrix} \begin{bmatrix} F_{512} & 0 \\ 0 & F_{512} \end{bmatrix} \begin{bmatrix} \text{even-odd} \\ \text{permutations} \end{bmatrix}$$

I_{512} is the identity matrix. D_{512} is the diagonal matrix with entries $(1, w, ..., w^{511})$. The two copies of F_{512} are what we expected, which use the 512^{th} root of unity, which is just $(w_{1024})^2$.

FFT Recursion

We reduced for F_N to $F_{N/2}$. So lets keep going to $F_{N/4}$. The two copies of F_{512} lead to four copies of F_{256}. This is the recursion.

$$\begin{bmatrix} F_{512} & 0 \\ 0 & F_{512} \end{bmatrix} = \begin{bmatrix} I_{256} & D_{256} & & \\ I_{256} & -D_{256} & & \\ & & I_{256} & D_{256} \\ & & I_{256} & -D_{256} \end{bmatrix} \begin{bmatrix} F_{256} & & & \\ & F_{256} & & \\ & & F_{256} & \\ & & & F_{256} \end{bmatrix} \begin{bmatrix} \text{pick 0,4,8...} \\ \text{pick 2,6,10...} \\ \text{pick 1,5,9...} \\ \text{pick 3,7,11...} \end{bmatrix}$$

We can count how many multiplications we have saved. When using the DFT we had $N^2 = (1024)^2$. This is about a million multiplications.

The final count for $N = 2^L$ is reduced from N^2 to $\frac{1}{2}NL$. The saving is therefore large.

When $N = 1024 = 2^{10}$, therefore $L = 10$. The original count of $(1024)(1024)$ is reduced to $(5)(1024)$.

FFT Complexity Reasoning

The reasoning behind $\frac{1}{2}NL$. There are L levels, going from $N = 2^L$ down to $N = 1$. Each level has $\frac{1}{2}N$ multiplications from the diagonal D to reassemble the half-size outputs.

This yields the final count $\frac{1}{2}NL$, which is $\frac{1}{2}N\log_2 N$.

The exact same idea gives rise to the fast inverse transform.

2.9 Data Flow Diagrams

The previous section demonstrates that the FFT consists of three parts: split the incoming signal into odd and even components; transform them separately and then perform a reconstruction. It is sometimes helpful to visualise this process using data flow diagrams. In this section a decimation in time example is given using data flow diagrams as an aid.

Decimation in Time

The radix-2 decimation in time algorithm starts by splitting $x[n]$ into two sequences:

$$f_e[n] = x[2n] \text{ (even samples) and } f_o[n] = x[2n+1] \text{ (odd samples), for } n = 0, \ldots, \frac{N}{2} - 1$$

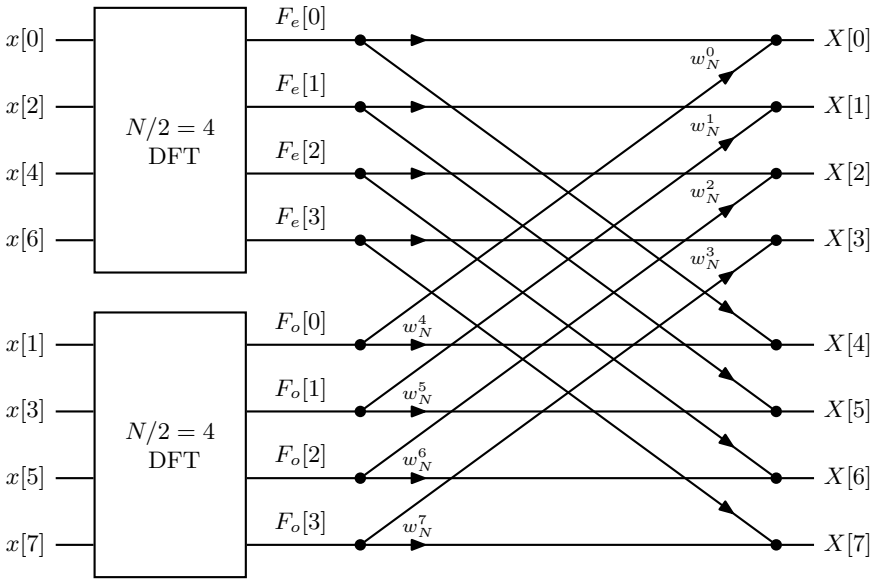
Therefore:

$$\begin{aligned} X[k] &= \sum_{n=0}^{N-1} x[n] w_N^{kn} \\ &= \sum_{m=0}^{N/2-1} x[2m] w_N^{k2m} + \sum_{m=0}^{N/2-1} x[2m+1] w_N^{k(2m+1)} \\ &= \sum_{m=0}^{N/2-1} f_e[m] w_{N/2}^{km} + w_N^k \sum_{m=0}^{N/2-1} f_o[m] w_{N/2}^{km} \quad \text{(remember } w_N^2 = w_{N/2}\text{)} \\ &= F_e[k] + w_N^k F_o[k] \quad \text{for } k = 0, 1, \cdots, N-1 \end{aligned}$$

where:

$$F_e[k] = \sum_{n=0}^{N/2-1} f_e[n] w_{N/2}^{kn} \quad \text{and} \quad F_o[k] \sum_{n=0}^{N/2-1} f_o[n] w_{N/2}^{kn}$$

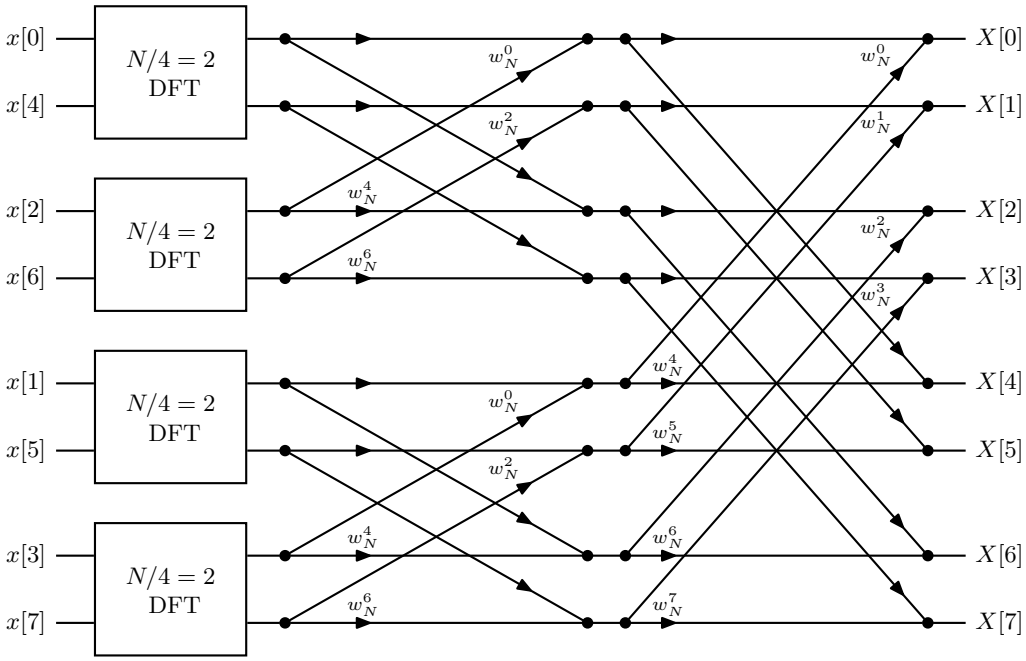
This result can now be visualised by a data flow diagram in the following way:



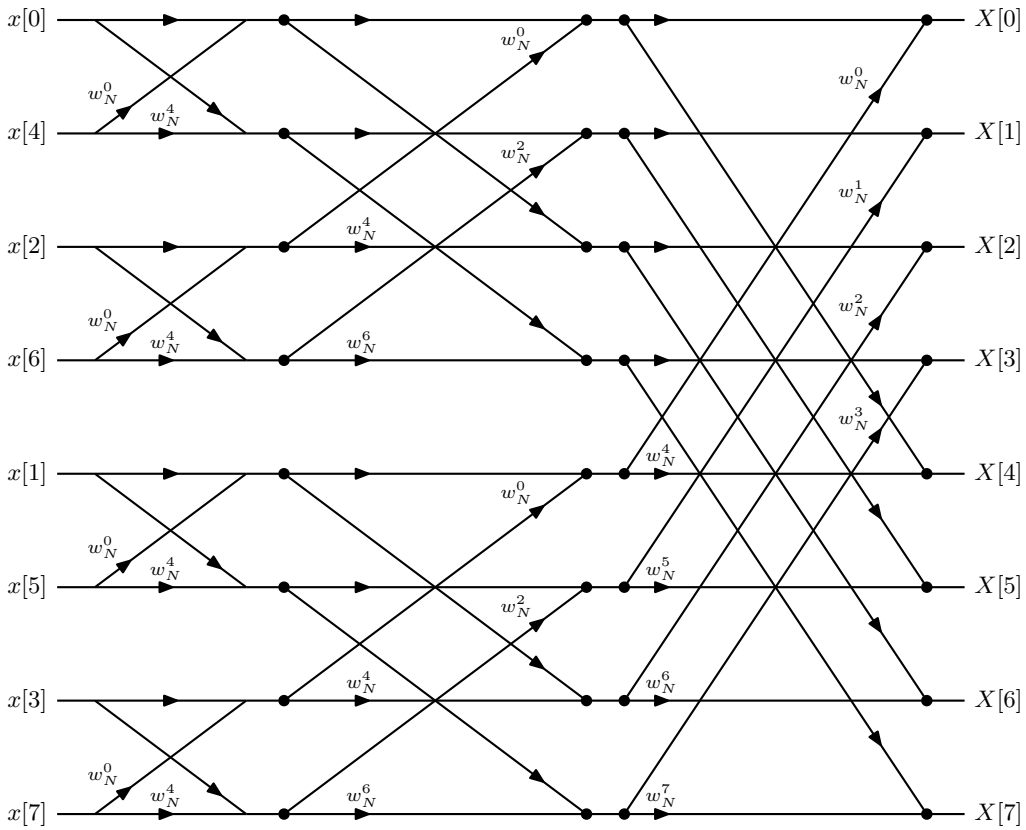
To calculate $X[4], X[5]X[6], X[7]$ recall that F[k] repeats every N/2 samples:

$$X[k] = F_e[k \bmod N/2] + w_N^k F_o[k \bmod N/2], \quad k = 0, \ldots, N-1.$$

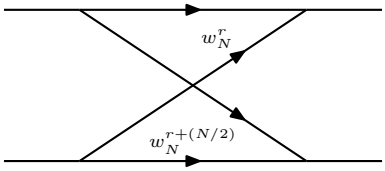
But why stop here... the process can now be repeated for the shorter (N/2) DFTs:



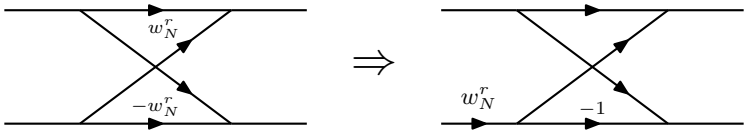
Again this process can be repeated for the shorter (N/4) DFTs:



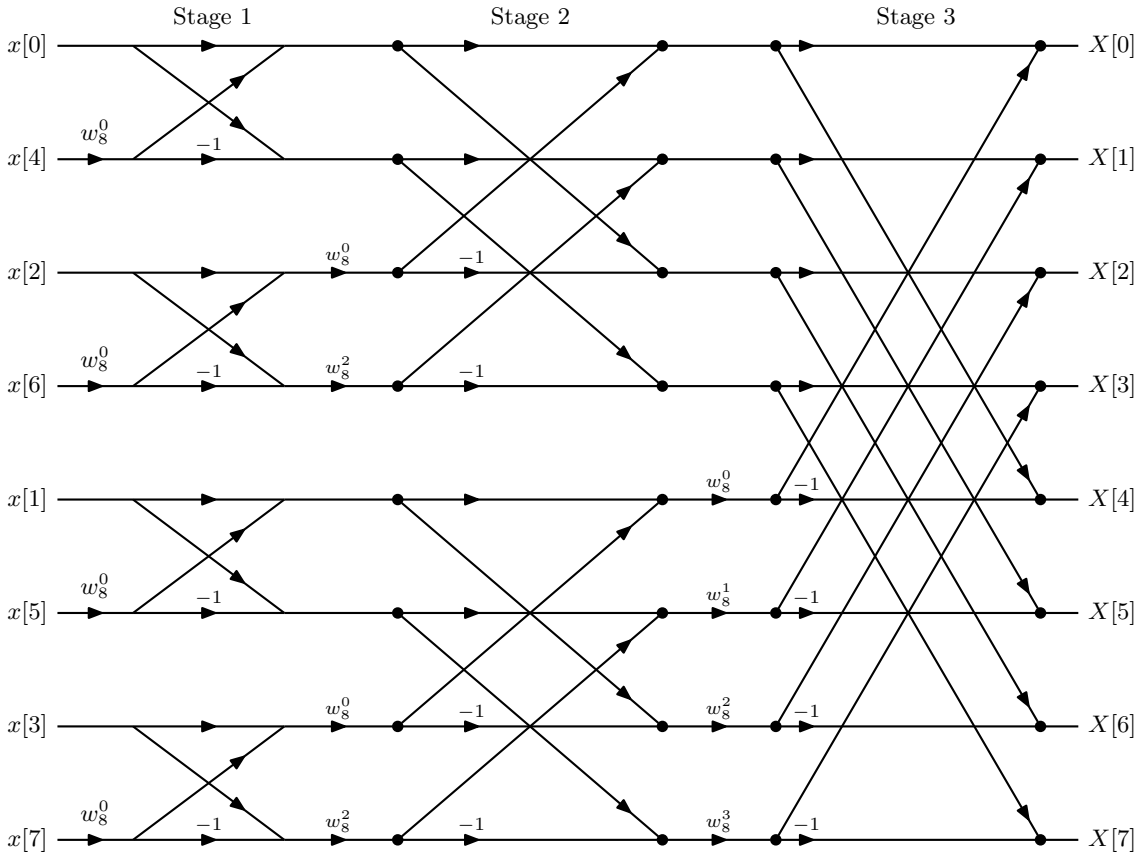
Now at each of these stages it is possible to see a pattern:



Notice that $w_N^{r+(N/2)} = w_N^r w_N^{N/2} = -w_N^r$ and, therefore, this 'butterfly' can be simplified in the following way:



It can be seen that this approach now saves a factor of N/2 complex multiplications at each stage. Thus the 8-point decimation in time FFT algorithm can be visualised in the following way:



FFT Computational Niceties

1. The FFT only contains $\frac{N}{2}\log_2 N$ complex multiplications.
2. The FFT input is ordered by bit-reversed addressing *(when the decimation happens in time).*
3. The FFT computation can be done 'in place' and, therefore, requires no extra storage.

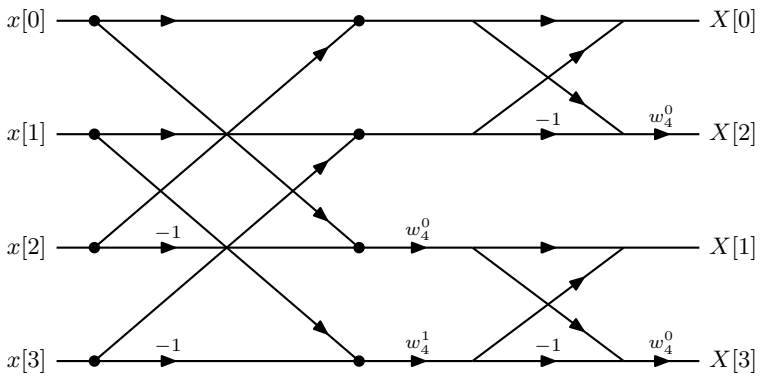
Decimation in Frequency

The radix-2 decimation in frequency algorithm splits the discrete Fourier transform into two parts:

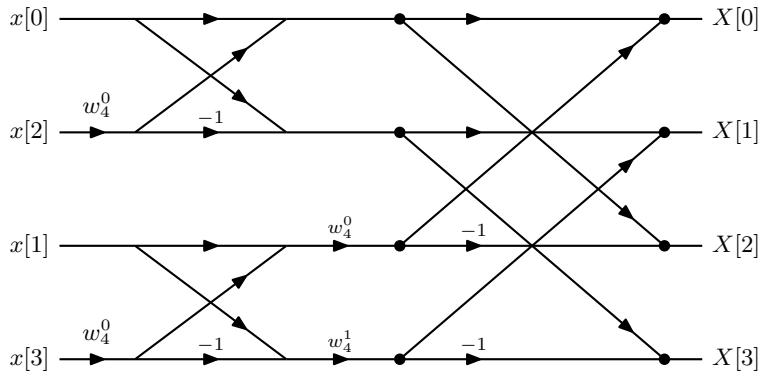
$$\begin{aligned} X[2k] &= \sum_{n=0}^{N-1} x[n] w_N^{2kn} \\ &= \sum_{n=0}^{N/2-1} x[n] w_N^{2kn} + \sum_{n=0}^{N/2-1} x[n+N/2] w_N^{2k(n+N/2)} \\ &= \sum_{n=0}^{N/2-1} x[n] w_N^{2kn} + \sum_{n=0}^{N/2-1} x[n+N/2] w_N^{2kn} \times 1 \\ &= \sum_{n=0}^{N/2-1} (x[n] + x[n+N/2]) w_{N/2}^{kn} \\ &= \sum_{n=0}^{N/2-1} g_1[n] w_{N/2}^{kn} \qquad g_1[n] = x[n] + x[n+N/2] \end{aligned}$$

$$\begin{aligned} X[2k+1] &= \sum_{n=0}^{N-1} x[n] w_N^{(2k+1)n} \\ &= \sum_{n=0}^{N/2-1} (x[n] + w_N^{N/2} x[n+N/2]) w_N^{(2k+1)n} \\ &= \sum_{n=0}^{N/2-1} ((x[n] - x[n+N/2]) w_N^n) w_{N/2}^{kn} \\ &= \sum_{n=0}^{N/2-1} g_2[n] w_{N/2}^{kn} \qquad g_2[n] = (x[n] - x[n+N/2]) w_N^n \end{aligned}$$

The signal flow graph follows as:



It is interesting to compare this approach against a radix-2 decimation in time 4-point FFT algorithm:



3 Module 3 - Convolution

3.1 Types of Convolution

Definitions:

- Linear Convolution: $y[n] = x[n] * h[n] \triangleq \sum_{k=-\infty}^{\infty} x[k]h[n-k]$
- Circular Convolution: $y[n] = x[n] \circledast_N h[n] \triangleq \sum_{k=0}^{N-1} x[k]h[(n-k)_{\text{mod } N}]$

DTFT

- Convolution $\rightarrow$ Product

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \Rightarrow Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$$

- Product $\rightarrow$ Circular Convolution $\div$ 2

$$y[n] = x[n]h[n] \Rightarrow Y(e^{j\omega}) = \frac{1}{2\pi}X(e^{j\omega}) \circledast_\pi H(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})H(e^{j(\omega-\theta)})d\theta$$

DFT

- Circular Convolution $\rightarrow$ Product

$$y[n] = x[n] \circledast_N h[n] = \sum_{k=0}^{N-1} x[k]h[(n-k)_{\text{mod } N}] \Rightarrow Y[k] = X[k]H[k]$$

- Product $\rightarrow$ Circular Convolution $\div$ N

$$y[n] = x[n]h[n] \Rightarrow Y[k] = \frac{1}{N}X[k] \circledast_N H[k]$$

3.2 Convolution Properties

Thankfully convolution obeys the normal arithmetic rules for multiplication.

1. Associative: $x[n] * (h[n] * v[n]) = (x[n] * h[n]) * v[n]$ therefore $x[n] * h[n] * v[n]$ is not ambiguous

   Proof: $\sum_{k=-\infty}^{\infty}\sum_{r=-\infty}^{\infty} x[n-k]h[k-r]v[r] \overset{\text{(i)}}{=} \sum_{p=-\infty}^{\infty}\sum_{q=-\infty}^{\infty} x[p]h[q-p]v[n-q]$
   (i) *substitute* $p = n-k$, $q = n-r$

2. Commutative: $x[n] * h[n] = h[n] * x[n]$

   Proof: $\sum_{k=-\infty}^{\infty} x[k]h[n-k] \overset{\text{(i)}}{=} \sum_{p=-\infty}^{\infty} x[n-p]h[p]$
   (i) *substitute* $p = n-k$

3. Distributive over addition: $x[n] * (\alpha h[n] + \beta v[n]) = (x[n] * \alpha h[n]) + (x[n] * \beta v[n])$

   Proof: $\sum_{k=-\infty}^{\infty} x[n-k](\alpha h[k] + \beta v[k]) = \alpha\sum_{k=-\infty}^{\infty} x[n-k]h[k] + \beta\sum_{k=-\infty}^{\infty} x[n-k]v[k]$

4. Identity: $x[n] * \delta = x[n]$

   Proof: $\sum_{k=-\infty}^{\infty} \delta[k]x[n-k] \overset{\text{(i)}}{=} x[n]$
   (i) *all terms are zero except when* $k = 0$

3.3 Calculating the Convolution

Step By Step Method

It is possible to compute convolution step by step:

-2	-1	0	1	2	k
	$x[-1]$	$x[0]$	$x[1]$	$x[2]$	$x[k]$
$h[-2]$	$h[-1]$	$h[0]$	$h[1]$		$h[k]$
	$h[1]$	$h[0]$	$h[-1]$	$h[-2]$	$h[-k]$
$\boldsymbol{y[0] =}$	$\boldsymbol{x[-1]h[1]+}$	$\boldsymbol{x[0]h[0]+}$	$\boldsymbol{x[1]h[-1]+}$	$\boldsymbol{x[2]h[-2]}$	
		$h[1]$	$h[0]$	$h[-1]$	$h[1-k]$
	$\boldsymbol{y[1] =}$	$\boldsymbol{x[0]h[1]+}$	$\boldsymbol{x[1]h[0] + x[2]h[-1]}$		
	$h[0]$	$h[-1]$	$h[-2]$		$h[-1-k]$
$\boldsymbol{y[-1] =}$	$\boldsymbol{x[-1]h[0]+}$	$\boldsymbol{x[0]h[-1] +}$	$\boldsymbol{x[1]h[-2]}$		
			$h[1]$	$h[0]$	$h[2-k]$
		$\boldsymbol{y[2] =}$	$\boldsymbol{x[1]h[1] + x[2]h[0]}$		
	$h[-1]$	$h[-2]$			$h[-2-k]$
$\boldsymbol{y[-2] =}$	$\boldsymbol{x[-1]h[-1] + x[0]h[-2]}$				
				$h[1]$	$h[3-k]$
			$\boldsymbol{y[3] =}$	$\boldsymbol{x[2]h[1]}$	
	$h[-2]$				$h[-3-k]$
$\boldsymbol{y[-3] =}$	$\boldsymbol{x[-1]h[-2]}$				

Simple Trick Method

This is a laborious process, however, there is a a very simple trick that can be used to calculate the convolution:

		-2	-1	0	1	2	k
			$x[-1]$	$x[0]$	$x[1]$	$x[2]$	$x[k]$
		$h[-2]$	$h[-1]$	$h[0]$	$h[1]$		$h[k]$
			$h[1]x[-1]$	$h[1]x[0]$	$h[1]x[1]$	$h[1]x[2]$	
		$h[0]x[-1]$	$h[0]x[0]$	$h[0]x[1]$	$h[0]x[2]$	$\times$	
	$h[-1]x[-1]$	$h[-1]x[0]$	$h[-1]x[1]$	$h[-1]x[2]$	$\times$		
$h[-2]x[-1]$	$h[-2]x[0]$	$h[-2]x[1]$	$h[-2]x[2]$	$\times$			
$y[-3]$	$y[-2]$	$y[-1]$	$y[0]$	$y[1]$	$y[2]$	$y[3]$	

Circulant Matrix Method

The last way to calculate the convolution is using a circulant matrix, where the circular convolution $y[n] = x[n] \circledast h[n]$ is the equivalent to:

$$\begin{bmatrix} x[0] & x[3] & x[2] & x[1] \\ x[1] & x[0] & x[3] & x[2] \\ x[2] & x[1] & x[0] & x[3] \\ x[3] & x[2] & x[1] & x[0] \end{bmatrix} \begin{bmatrix} h[0] \\ h[1] \\ h[2] \\ h[3] \end{bmatrix} = \begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ y[3] \end{bmatrix}$$

Where $h[n]$ and $x[n]$ are made up of 4 samples.

Remember you can zero pad the signals to get the linear convolution:

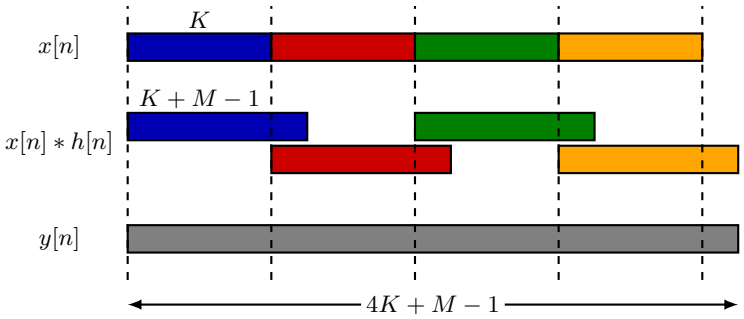
$$x[k] = \{x[-1], x[0], x[1], x[2], 0, 0, 0\}$$
$$h[k] = \{h[-2], h[-1], h[0], h[1]0, 0, 0\}$$

$$\begin{bmatrix} x[-1] & 0 & 0 & 0 & x[2] & x[1] & x[0] \\ x[0] & x[-1] & 0 & 0 & 0 & x[2] & x[1] \\ x[1] & x[0] & x[-1] & 0 & 0 & 0 & x[2] \\ x[2] & x[1] & x[0] & x[-1] & 0 & 0 & 0 \\ 0 & x[2] & x[1] & x[0] & x[-1] & 0 & 0 \\ 0 & 0 & x[2] & x[1] & x[0] & x[-1] & 0 \\ 0 & 0 & 0 & x[2] & x[1] & x[0] & x[-1] \end{bmatrix} \begin{bmatrix} h[-2] \\ h[-1] \\ h[0] \\ h[1] \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} y[-3] \\ y[-2] \\ y[-1] \\ y[0] \\ y[1] \\ y[2] \\ y[3] \end{bmatrix}$$

3.4 Overlap Add

If N is very large:

1. chop $x[n]$ into $\frac{N}{K}$ chunks of length K
2. convolve each chunk with $h[n]$
3. add up the results



Each output chunk is of length $K + M - 1$ and overlaps the next chunk

Number of operations: $\approx \frac{N}{K} \times 8(M + K) \log_2(M + K)$

Computational saving if $\approx 100 < M \ll K \ll N$

Other advantages:

(a) Is able to handle $N = \infty$

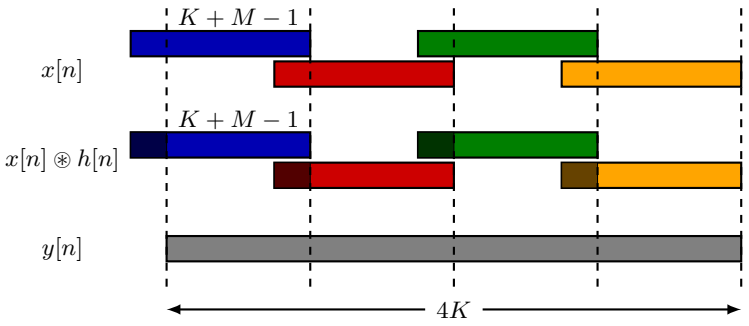
(b) The DFT needed is shorter *(if the DFT is used to perform circular convolution which is more computationally efficient)*

(c) Can calculate $y[0]$ as soon as $x[K-1]$ has been read, however, there is a algorithmic delay of $K-1$ samples.

3.5 Overlap Save

If N is very large:

1. chop $x[n]$ into $\frac{N}{K}$ overlapping chunks of length $K+M-1$
2. $\circledast_{K+M-1}$ each chunk with $h[n]$
3. discard first $M-1$ from each chunk
4. concatenate to make y[n]



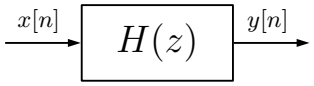
The first $M-1$ points of each output chunk are invalid.

Number of operations: slightly less than overlap-add due to the fact that no additions are needed to create $y[n]$. The advantages are the same as with overlap add, however, much less popular than overlap-add which is strange.

4 Module 4 - Digital Filters: Implementation and Design

4.1 Filters

Difference Equations



A difference equation is way of describing most useful LTI systems:

$$
\begin{aligned}
y[n] &= \textstyle\sum_{r=0}^{M} b[r]x[n-r] - \sum_{r=1}^{N} a[r]y[n-r] \\
\Leftrightarrow \textstyle\sum_{r=0}^{N} a[r]y[n-r] &= \textstyle\sum_{r=0}^{M} b[r]x[n-r], \qquad \text{where } a[0]=1 \\
\Leftrightarrow a[n] * y[n] &= b[n] * x[n] \\
\Leftrightarrow Y(z) &= \frac{B(z)}{A(z)}X(z) \\
\Leftrightarrow Y(e^{j\omega}) &= \frac{B(e^{j\omega})}{A(e^{j\omega})}X(e^{j\omega})
\end{aligned}
$$

(a) Always causal

(b) Order of system is $\max(M, N)$, the highest r with $a[r] \neq 0$ or $b[r] \neq 0$

(c) We assume that $a[0] = 1$; if not, divide $A(z)$ and $B(z)$ by $a[0]$

(d) Filter is BIBO stable if roots of A(z) all lie within the unit circle

Note the negative sign in the first equation. In [4] the authors reverse the sign of the $a[n]$, however, this is actually a bad idea.

Recursive and Non Recursive

A digital system in general is described by a difference equation. Here are two examples:

$$
\begin{aligned}
y[n] &= x[n] + x[n-1] + \cdots + x[n-50] \\
y[n] &= y[n-1] + x[n] - x[n-51]
\end{aligned}
$$

The first one is a non recursive difference equation while the second one is recursive. The important point to notice is that both these equations implement the same system.

Proof

$$
\begin{aligned}
y[n] &= x[n] + x[n-1] + \cdots + x[n-50] && \text{(a)} \\
y[n-1] &= x[n-1] + x[n-2] + \cdots + x[n-51] && \text{(b)} \\
y[n] - y[n-1] &= x[n] - x[n-51] && \text{(a) - (b)} \\
y[n] &= y[n-1] + x[n] - x[n-51] && \text{Requires less additions than (a)}
\end{aligned}
$$

Note:

(a) FIR filters are normally implemented non recursively but can be implemented recursively.

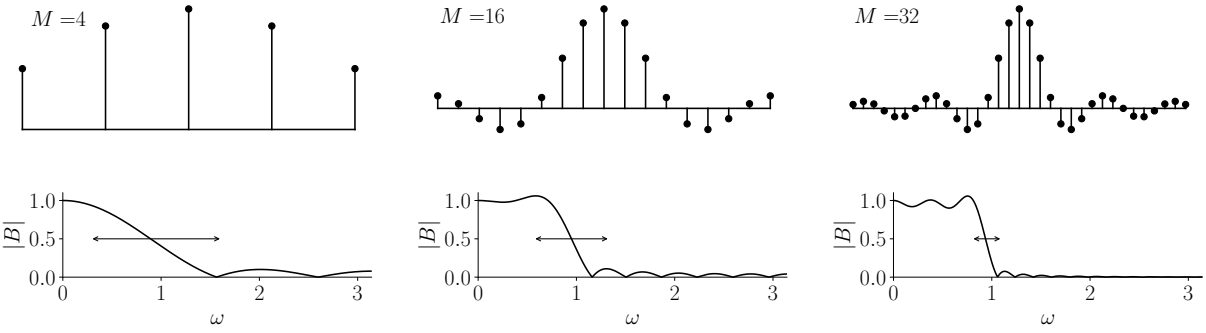
(b) IIR filters can only be implemented recursively in practice because an infinite number of coefficients would be required to realise them non recursively (recall: $y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$).

FIR Filters

A Finite Impulse Response (FIR) filter is one where $A(z) = 1$, and therefore, $Y(z) = B(z)X(z)$.

The impulse response being $b[n]$ with a length of $M+1$ and the frequency response being $B(e^{j\omega})$ which is just the DTFT of $b[n]$.

Note: If M is small then the frequency response only contains low 'quefrencies' *('quefrencies' is not a typographical error)*. Also if $b[n]$ is symmetric then $H(e^{j\omega})e^{\frac{jM\omega}{2}}$ consists of $\frac{M}{2}$ cosine waves plus a constant.



Rule of thumb: The fastest possible transition is $\Delta\omega \leq \frac{2\pi}{M}$ (the marked double arrow)

FIR Symmetries

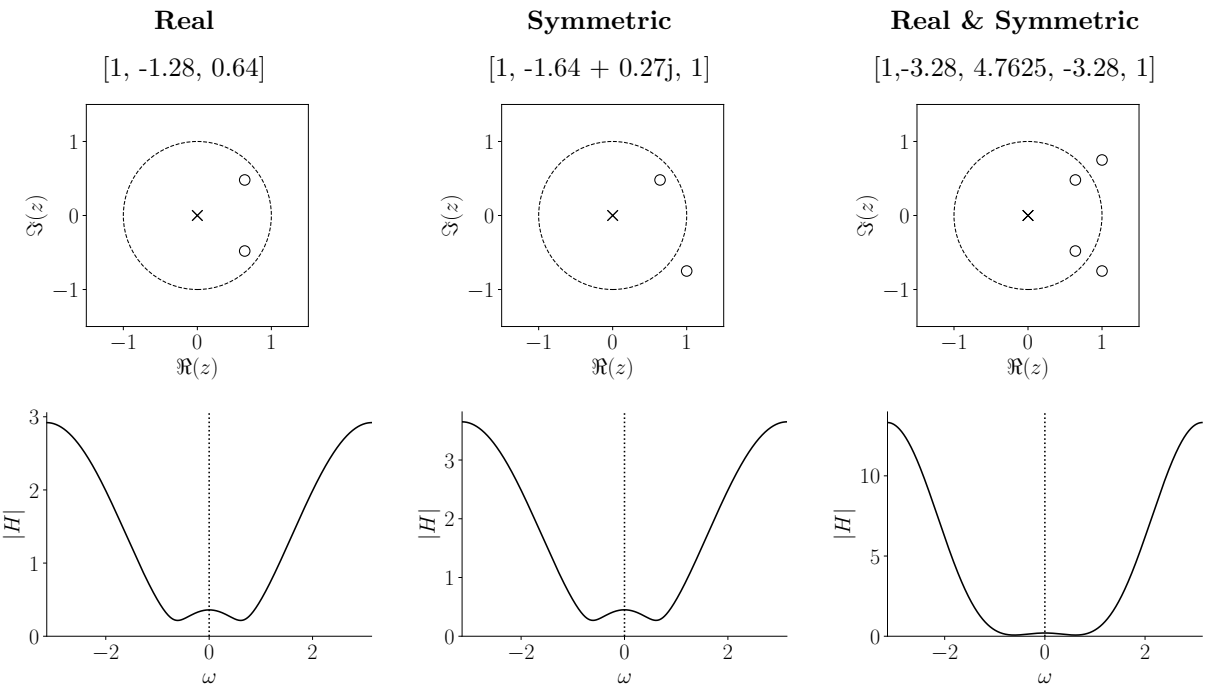
$B(e^{j\omega})$ is determined by the zeros of $z^M B(z) = \sum_{r=0}^{M} b[M-r]z^r$

It is advantageous to note some symmetric properties:

Real $b[n] \Rightarrow$ conjugate zero pairs: $z \Rightarrow z^*$

Symmetric: $b[n] = b[M-n] \Rightarrow$ reciprocal zero pairs: $z \Rightarrow z^{-1}$

Real & Symmetric $b[n] \Rightarrow$ conjugate and reciprocal groups of four (else pairs on the real axis)



IIR Frequency Response

An Infinite Impulse Response (IIR) filter is one where $Y(z) = \dfrac{B(z)}{A(z)}X(z)$.

Factorise

$$H(e^{j\omega}) = \frac{B(z)}{A(z)} = \frac{b[0]z^{-M}\prod_{i=1}^{M}(1-q_i z^{-1})}{\prod_{i=1}^{N}(1-p_i z^{-1})}$$

The poles p_i are the roots of $A(z)$ and $B(z)$. The zeros q_i are the zeros of $H(z)$. Note that there are additional $N-M$ zeros at the origin *(which affects only the phase)*

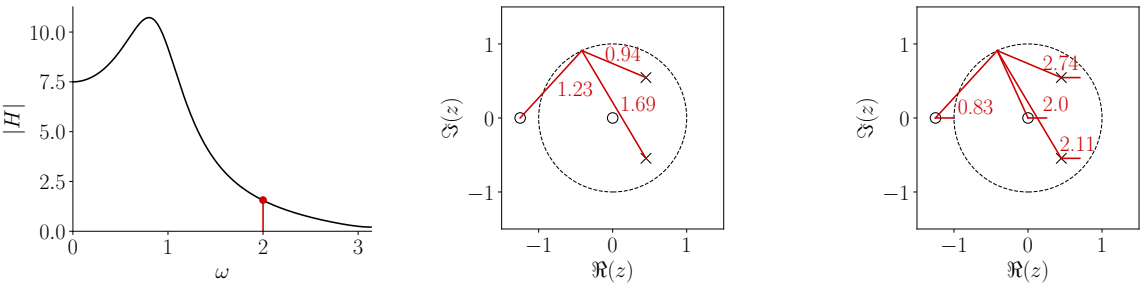
$$|H(e^{j\omega})| = \frac{|b[0]||z^{-M}|\prod_{i=1}^{M}|1-q_i z^{-1}|}{|z^{-N}|\prod_{i=1}^{N}|1-p_i z^{-1}|} \quad \text{for } z = e^{j\omega}$$

Example

$$H(z) = \frac{2+2.5z^{-1}}{1-0.9z^{-1}+0.5^{-2}} = \frac{2(1+1.25z^{-1})}{(1-(0.45-0.55j)z^{-1})(1-(0.45+0.55j)z^{-1})}$$

$$\text{At } \omega = 2: \quad |H(e^{j\omega})| = \frac{2*1.23}{1.69*0.94} = 1.6$$

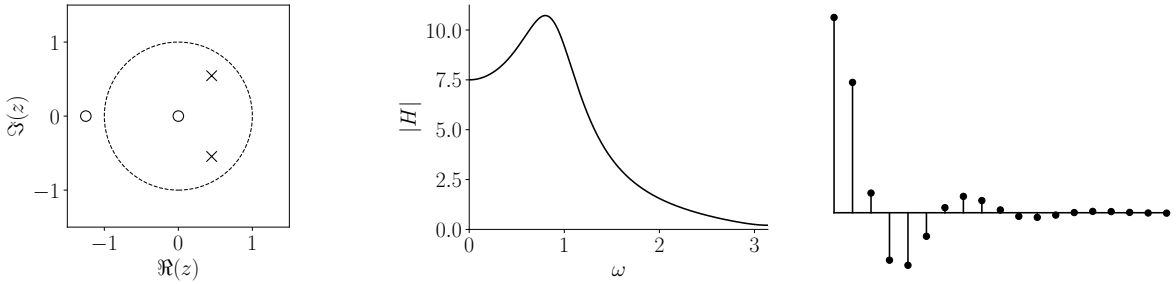
$$\angle H(e^{j\omega}) = (0.83+2.0) - (2.11+2.74) = -2.02$$



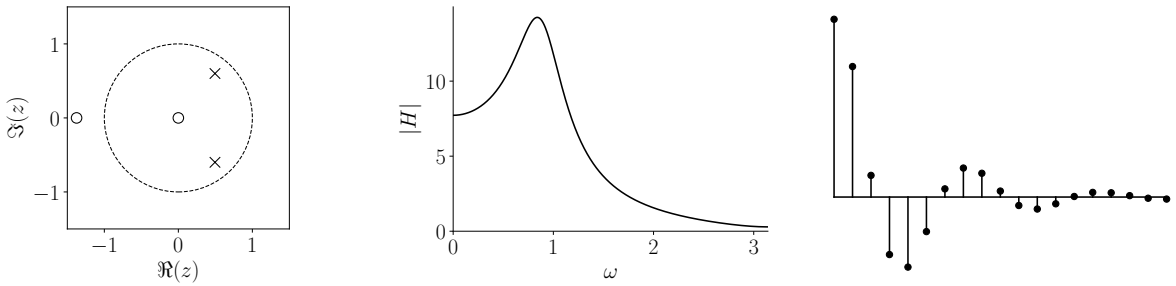
Scaling z

Given the filter $H(z)$ it is possible to form a new filter $H_S(z) = H\left(\frac{z}{\alpha}\right)$, which is equivalent to multiplying $a[n]$ and $b[n]$ by α^n.

$$\text{Example: } H(z) = \frac{2+2.5z^{-1}}{1-0.9z^{-1}+0.5^{-2}}$$



$$\text{Scale z: } H_S(z) = H\Big(\frac{z}{1.1}\Big) = \frac{2 + 2.75z^{-1}}{1 - 0.99z^{-1} + 0.605^{-2}}$$



Pole and zero positions are multiplied by $\alpha, \alpha > 1 \Rightarrow$ peaks sharpened.

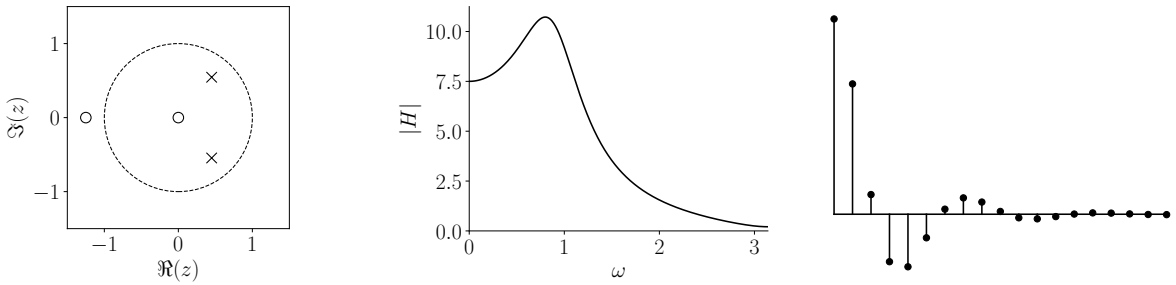
Pole at $z = p$ gives peak bandwidth $\approx 2|\log|p|| \approx 2(1 - |p|)$

For pole near unit circle, decrease bandwidth by $\approx 2\log\alpha$

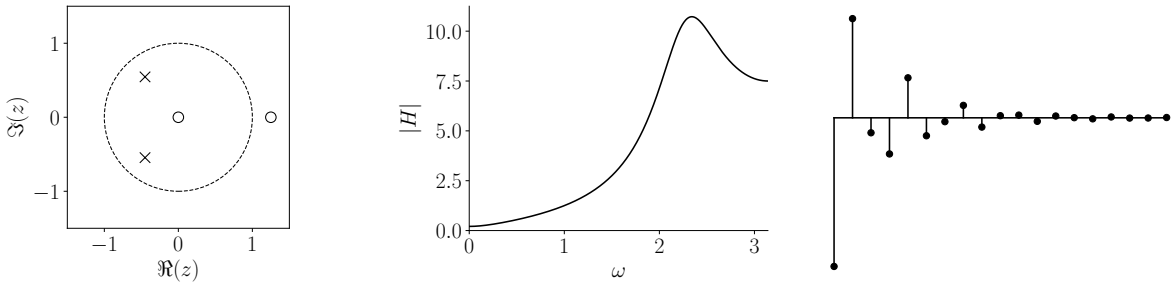
Negating z

Given the filter $H(z)$ it is possible to form a new filter $H_R(z) = H(-z)$, which is equivalent to negating all odd power of z, that is, negating alternate $a[n]$ and $b[n]$.

$$\text{Example: } H(z) = \frac{2 + 2.5z^{-1}}{1 - 0.9z^{-1} + 0.5^{-2}}$$



$$\text{Negate z: } H_R(z) = H(-z) = \frac{2 - 2.5z^{-1}}{1 + 0.9z^{-1} + 0.5^{-2}} \qquad \textit{Negate odd coefficients}$$

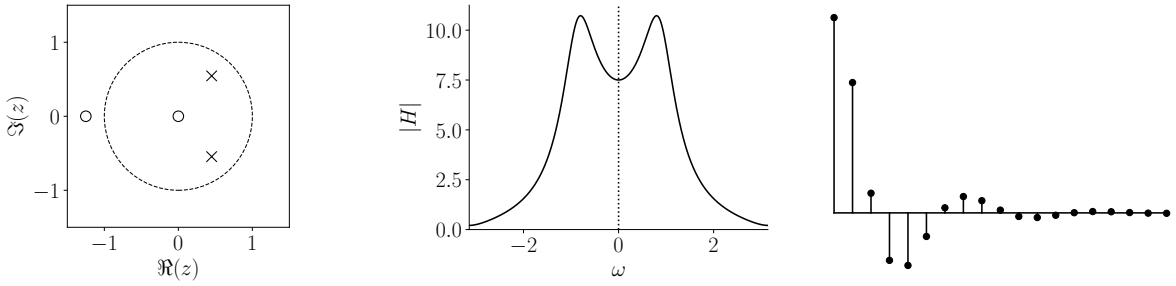


Pole and zero positions are negated, response is flipped and conjugated.

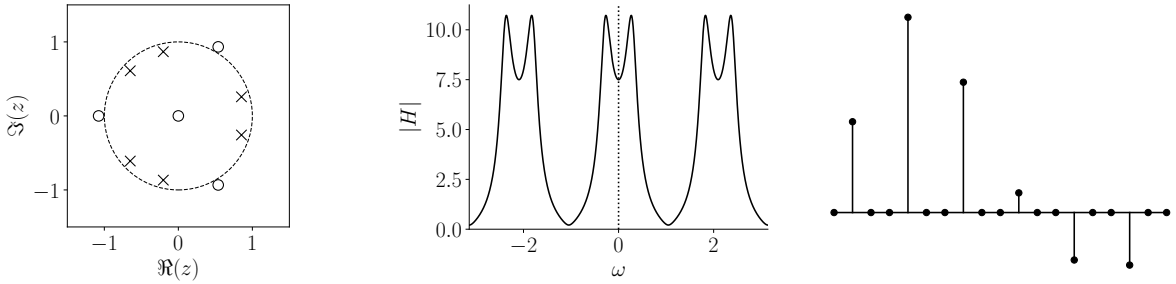
Cubing z

Given the filter $H(z)$ it is possible to form a new filter $H_C(z) = H(z^3)$, which is equivalent to inserting two zeros between each $a[n]$ and $b[n]$ term.

$$\text{Example: } H(z) = \frac{2 + 2.5z^{-1}}{1 - 0.9z^{-1} + 0.5^{-2}}$$



$$\text{Cube z: } H_c(z) = H(z^3) = \frac{2 + 2.5z^{-3}}{1 - 0.9z^{-3} + 0.5^{-6}} \qquad \textit{Insert 2 zeros between coefficients}$$



Pole and zero positions are replicated. Magnitude response is also replicated.

Group Delay

The group delay is defined $\tau_H(e^{jw}) = -\frac{d\angle H(e^{j\omega})}{d\omega}$ = delay of the modulation envelope.

One trick to get at the phase is: $\ln H(e^{j\omega}) = \ln|H(e^{j\omega})| + j\angle H(e^{j\omega})$

$$\tau_H = \frac{-d\big(\Im\big(\ln H(e^{j\omega})\big)\big)}{d\omega} = \Im\Big(\frac{-1}{H(e^{j\omega})}\frac{dH(e^{j\omega})}{d\omega}\Big) = \Re\Big(\frac{-z}{H(z)}\frac{dH(z)}{dz}\Big)\Big|_{z=e^{j\omega}}$$

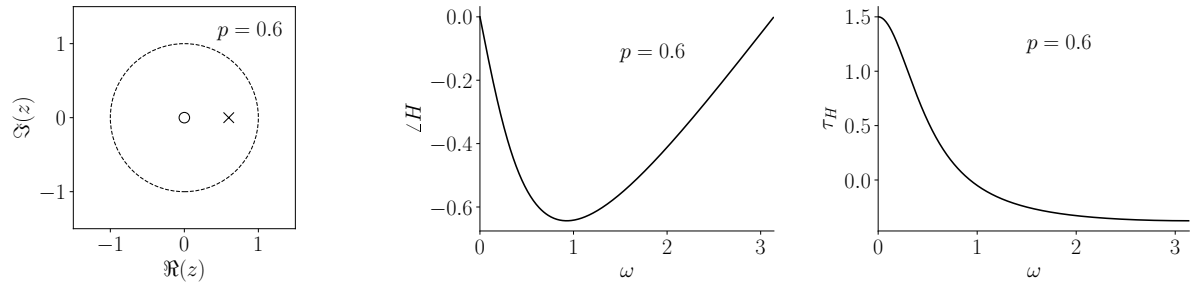
$$H(e^{j\omega}) = \sum_{n=0}^{\infty} h[n]e^{-jn\omega} = \mathscr{F}(h[n]), \qquad \text{where } \mathscr{F} \text{ denotes the DTFT.}$$

$$\frac{dH(e^{j\omega})}{d\omega} = \sum_{n=0}^{\infty} -jnh[n]e^{-jn\omega} = -j\mathscr{F}(nh[n])$$

$$\tau_H = \Im\Big(\frac{-1}{H(e^{j\omega})}\frac{dH(e^{j\omega})}{d\omega}\Big) = \Re\Big(\frac{\mathscr{F}(nh[n])}{\mathscr{F}(h[n])}\Big) = \Im\Big(j\frac{\mathscr{F}(nh[n])}{\mathscr{F}(h[n])}\Big)$$

Example:

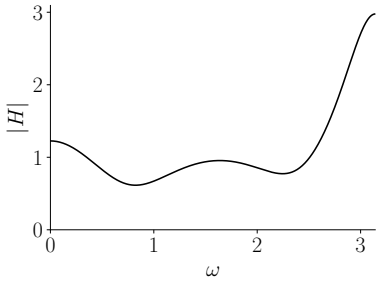
$$H(z) = \frac{1}{1 - pz^{-1}} \Rightarrow \tau_H = -\tau_{[1-p]} = -\Re\Big(\frac{-pe^{-j\omega}}{1 - pe^{-j\omega}}\Big)$$

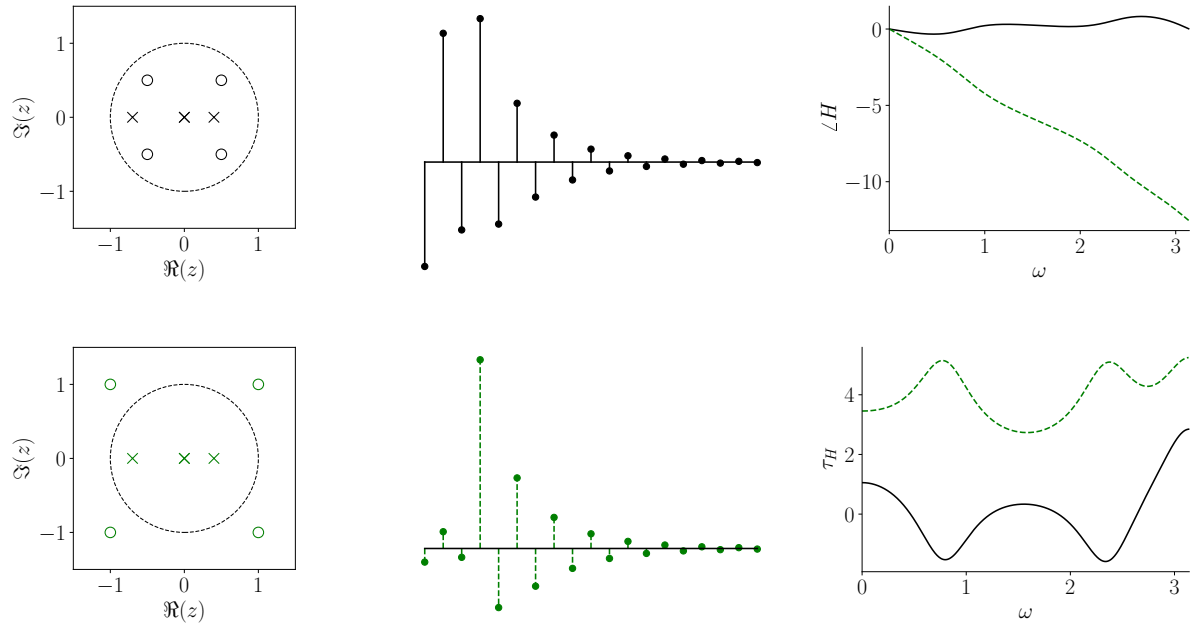


Minimum Phase

Average group delay over ω equals the (#poles - #zeros) within the unit circle where zeros on the unit circle count for $\frac{1}{2}$

Reflecting an interior zero to the exterior multiplies $|H(e^{j\omega})|$ by a constant and **increases the average group** delay by 1 sample.





A filter with all its zeros inside the unit circle is a minimum phase filter:

1. The lowest possible group delay for a given magnitude response
2. The energy in $h[n]$ is concentrated towards $n = 0$

In contrast a filter with all its zeros outside the unit circle is a maximum phase filter

Linear Phase Filters

The phase of a linear phase filter is: $\angle H(e^{j\omega}) = \theta_0 - \alpha$

This is equivalent to the group delay being constant: $\tau_H = -\frac{d\angle H(e^{j\omega})}{d\omega} = \alpha$

A filter has linear phase, if and only if, its impulse response $h[n]$ is symmetric or antisymmetric:

$$h[n] = h[M-n] \quad \forall n \qquad \text{or else} \qquad h[n] = -h[M-n] \quad \forall n$$

M can be even ($\Rightarrow \ni$ mid point) or odd ($\Rightarrow \not\ni$ mid point)

Important: This is not the same symmetry that is needed to make the signal real in the frequency domain, which is when $h[n] = h[-n]$ where $M = N - 1$.

4.2 FIR Digital Filter Design

The frequency response for any BIBO stable filter can be denoted by $H(e^{j\omega})$. Since $H(e^{j\omega})$ is a periodic function with period 2π, it can be expressed as a Fourier series:

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h[n]e^{-j\omega n}$$

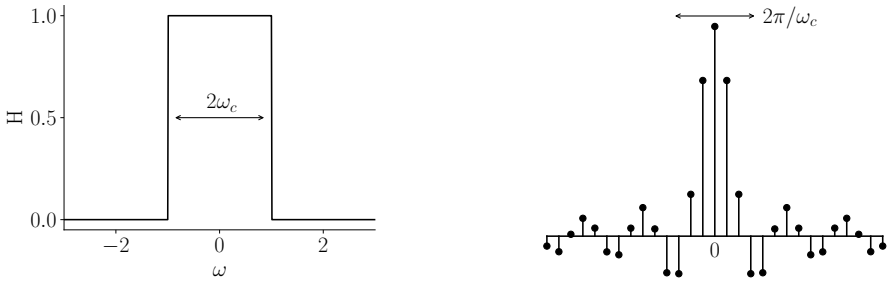
Where the Fourier coefficients $h[n]$ are equal to the impulse response samples which are given by:

$$h[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\omega})e^{j\omega n}d\omega, \quad \infty < n < \infty$$

Thus given we know $H(e^{j\omega})$, we can compute $h[n]$ using the IDTFT.

Example: Impulse Response of Ideal Low-pass Filter

$$H(e^{j\omega}) = \begin{cases} 1, & |\omega| \le \omega_c \\ 0, & |\omega| > \omega_c \end{cases} \quad \Leftrightarrow \quad h[n] = \frac{sin(\omega_c n)}{\pi n}$$

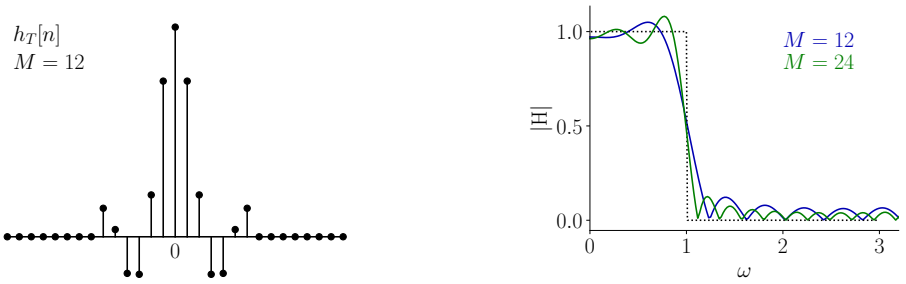


Note: Width in ω is $2\omega_c$, width in n is $2\pi/\omega_c$ where the product is always 4π.

The problem is $h[n]$ is infinite and non-causal which makes it unreliable To solve this problem we multiply $h[n]$ by a window (truncating the coefficients) and then shift this new windowed signal to the right appropriately.

Windowing Method

We therefore truncate $h[n]$ to $\pm\frac{M}{2}$ to make it finite, so $h_T[n]$ is now of length $M+1$.



It would then be normal to delay by $h_T[n]$ by $\frac{M}{2}$ to make it causal. Which multiplies $H(e^{j\omega})$ by $e^{j\frac{M}{2}\omega}$.

Gibbs Phenomenon

This type of truncation is optimal in terms of mean square error. If we define the mean squared error in the frequency domain to be:

$$\begin{aligned} E &= \frac{1}{2\pi}\int_{-\pi}^{\pi} |H(e^{j\omega}) - H_T(e^{j\omega})|^2 d\omega \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} |H(e^{j\omega}) - \textstyle\sum_{-\frac{M}{2}}^{\frac{M}{2}} h_T[n]e^{-j\omega n}|^2 d\omega \end{aligned}$$

In this case E is minimised when $h_T[n] = h[n]$. *Proof:* $E = \sum_{-\pi}^{\pi} |h[n] - h_T[n]|^2 + \sum_{|n|>\pi} |h[n]|^2$

From this result it would be easy to assume that a rectangular window is therefore the best choice, however, this is almost never true due to the fact that no matter how large you make M, there will, always be a 9% overshoot at any discontinuity. This behaviour is often referred to as 'Gibbs phenomenon'. Thus other windows are often used over the rectangular window to introduce a smooth transition in the filter specification and thus eliminate 'Gibbs phenomenon'.

Dirichlet Function

The reason behind Gibbs phenomenon can be explained by considering the truncation process which can be expressed as the multiplication of $h[n]$ by a rectangular window $w[n] = \delta_{-\frac{M}{2} \leq n \leq \frac{M}{2}}$.

This can be alternatively expressed as a circular convolution due to the modulation theorem.

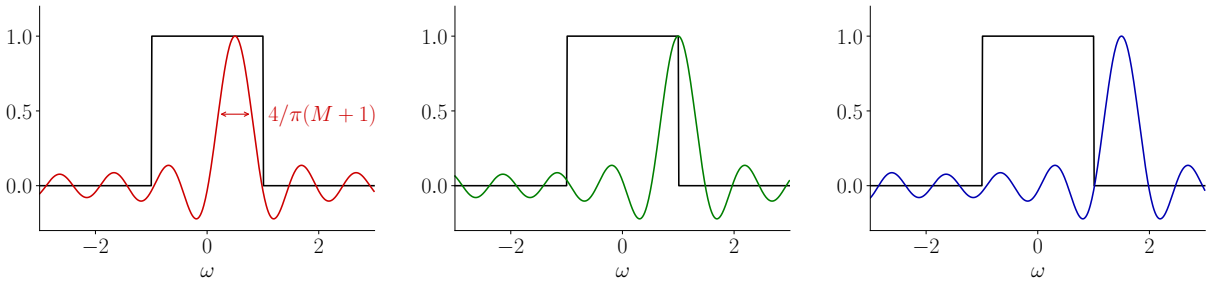
$$H_{M+1}(e^{j\omega}) = \frac{1}{2\pi} H(e^{j\omega}) \circledast W(e^{j\omega})$$

where

$$W(e^{j\omega}) = \textstyle\sum_{-\frac{M}{2}}^{\frac{M}{2}} e^{-j\omega n} \overset{\text{(i)}}{=} 1 + 2\sum_{1}^{\frac{M}{2}} \cos(n\omega) \overset{\text{(ii)}}{=} \frac{\sin([M+1]\omega/2)}{\sin(0.5\omega)}$$

The proof being: (i) $e^{-j\omega(-n)} + e^{-j\omega(+n)} = 2\cos(n\omega)$ (ii) Sum of a geometric progression

This has the effect of convolving the ideal frequency response with an aliased sinc function, which is normally called the 'Dirichlet function'.



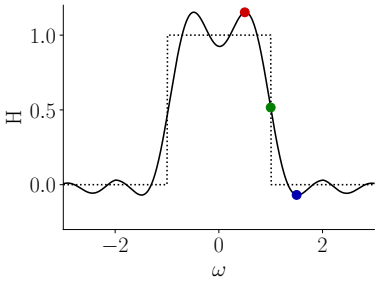
These figures give an illustration of the effect of windowing in the frequency domain.

Provided that $\frac{4\pi}{M+1} \ll 2\omega_c \Leftrightarrow M+1 \gg \frac{2\pi}{\omega}$:

Passband ripple: $\Delta\omega \approx \frac{4\pi}{M+1}$, stopband $\frac{2\pi}{M+1}$

Transition peak-to-peak: $\Delta\omega \approx \frac{4\pi}{M+1}$

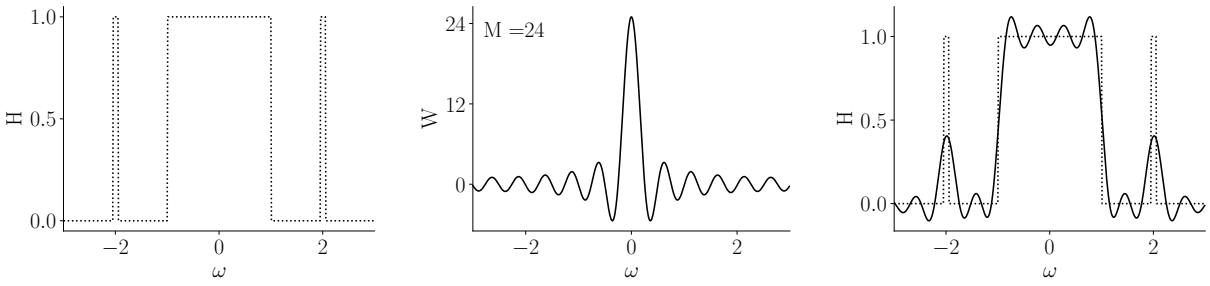
Transition gradient: $\frac{d|H|}{d\omega}\Big|_{\omega=\omega_c} \approx \frac{M+1}{2\pi}$



Window Relationships

Relationships when you multiply an impulse response $h[n]$ by a window $w[n]$ that is $M+1$ long

$$H_{M+1}(e^{j\omega}) = \frac{1}{2\pi} H(e^{j\omega}) \circledast W(e^{j\omega})$$



(a) Passband gain $\approx w[n]$; peak $\approx \frac{w[0]}{2} + \frac{0.5}{2\pi}\int_{\text{mainlobe}} W(e^{j\omega})d\omega$

rectangular window: passband gain = 1; peak gain = 1.09

(b) Transition bandwidth, $\Delta\omega$ = width of the main lobe
Transition amplitude, ΔH = integral of main lobe divide by 2π

rectangular window: $\Delta\omega = \frac{4\pi}{M+1}, \Delta H \approx 1.18$

(c) Stopband gain is an integral over oscillating sidelobes of $W(e^{j\omega})$

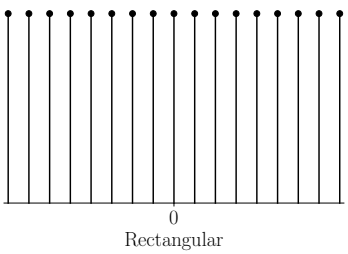
rectangular window: $|\min H(e^{j\omega})| = 0.09 \ll |\min W(e^{j\omega})| = \frac{M+1}{1.5\pi}$

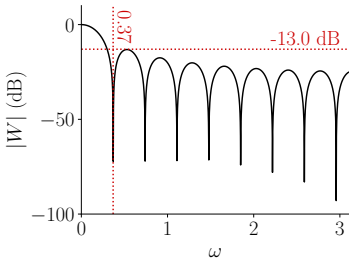
(d) The features that are narrower than the main lobe will be broadened and attenuated

Common Windows

Rectangular:

$w[n] = \delta_{-\frac{M}{2} \leq n \leq \frac{M}{2}}$ Main lobe width: $4\pi/(M+1)$
Relative sidelobe level: 13.0dB
Never use

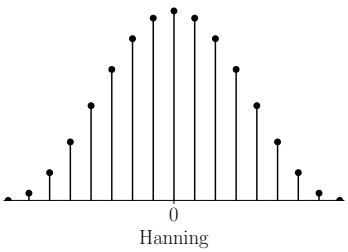


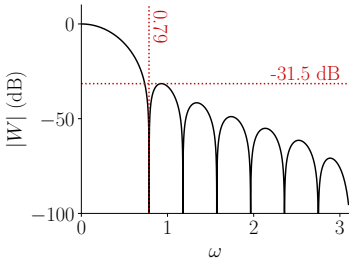


Hanning:

$w[n] = 0.5 + 0.5\cos\frac{2\pi n}{M+1}$

Main lobe width: $8\pi/(M+1)$
Relative sidelobe level: 31.5dB
Rapid sidelobe decay

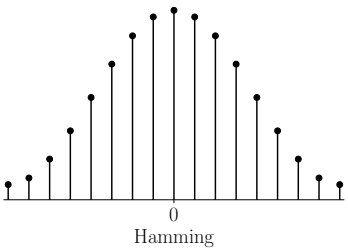


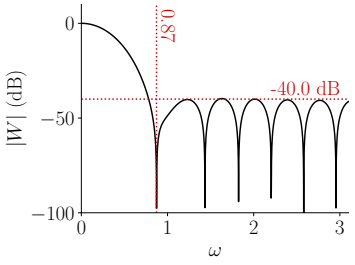


Hamming:

$w[n] = 0.54 + 0.46\cos\frac{2\pi n}{M+1}$

Main lobe width: $4\pi/(M+1)$
Relative sidelobe level: 40.0dB
Best peak sidelobe

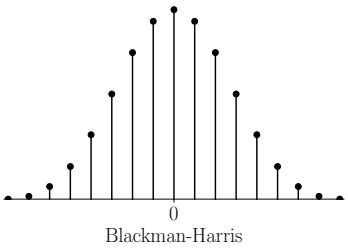


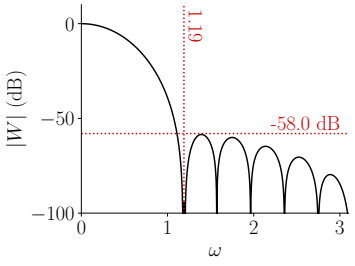


Blackman-Harris (3-term):

$w[n] = 0.42 + 0.5\cos\frac{2\pi n}{M+1} + 0.08\cos\frac{4\pi n}{M+1}$

Main lobe width: $12\pi/(M+1)$
Relative sidelobe level: 58.0dB
Best peak sidelobe





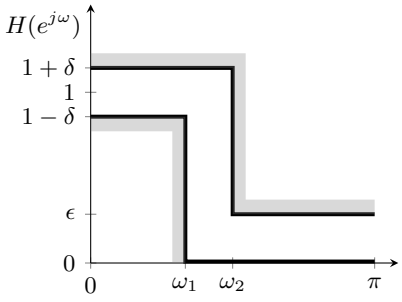
FIR Filter Order Estimation

Several formulae estimate the required order M of a filter.

One that is often used is the 'Fred Harris approximation':

$$M \approx \frac{A}{20(\omega_2 - \omega_1)/2\pi}, \quad \text{where } A = 20\log_{10}(\epsilon)$$

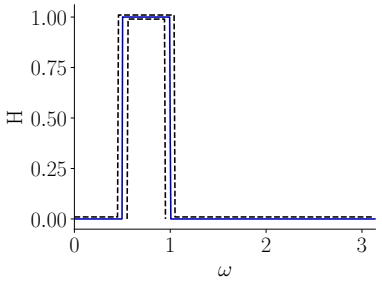
This is, of course, of only approximate.



For example:

Specifications:

Bandpass: $0.5 \leq |\omega| \leq 1$
Transition bandwidth: $\Delta\omega = 0.1$
Ripple: $\delta = \epsilon = 0.01$
$20\log_{10}(\epsilon) = -40$ dB
$20\log_{10}(1 + 0.01) = 0.09$ dB



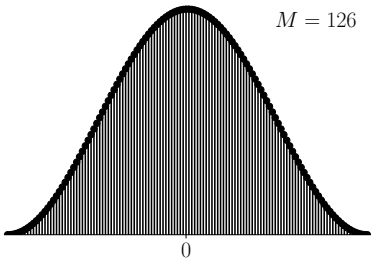
Order:

$$M \approx \frac{A}{20(\omega_2 - \omega_1)/2\pi} = \frac{40}{20(0.1)/2\pi} = 126$$

Ideal Impulse Response:

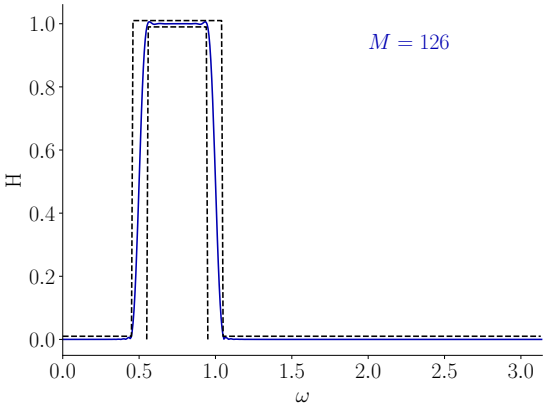
Difference of two low-pass filters:

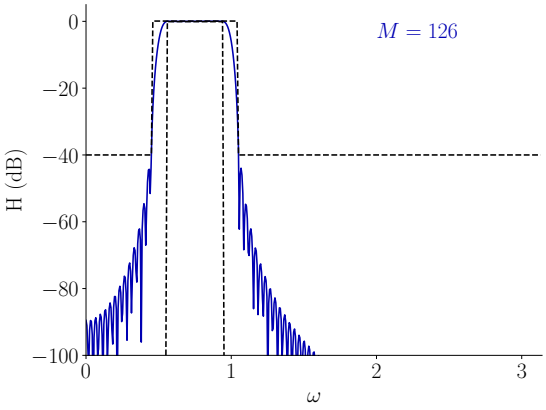
$h[n] = \frac{\sin \omega_2 n}{\pi n} - \frac{\sin \omega_1 n}{\pi n}$



A Hanning window will be used where M is set to 126

The figures below show the result of this type of filter design.





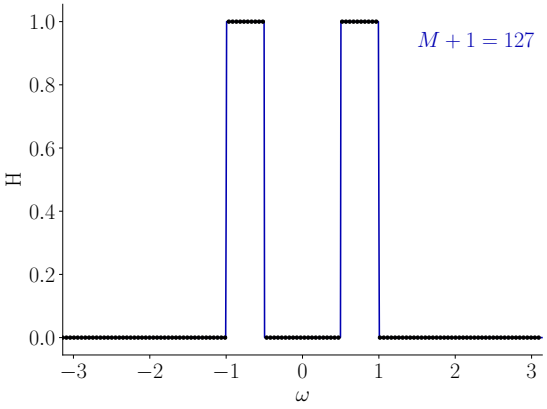
Frequency Sampling

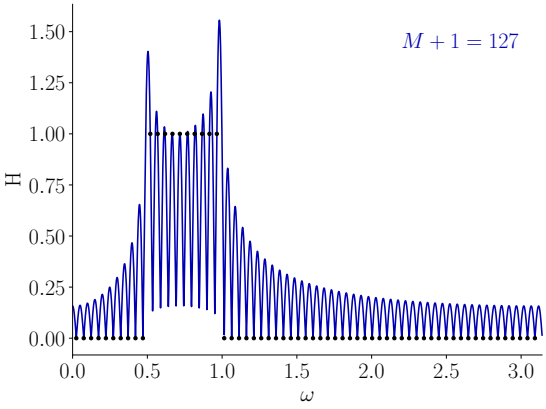
In the windowing method we truncate $h[n]$ to $\pm\frac{M}{2}$ to make it finite, however, another approach could be to take the IDFT of $M+1$ equally spaced samples of $H(e^{j\omega})$.

On the surface, this appears to be a good idea due to the fact that it has the advantage of giving an exact match at the sample points.

However, it unfortunately has one big disadvantage which is that the intermediate approximation is poor if the spectrum varies rapidly.

This can be seen if we design the previous filter using the frequency sampling method with the same value of M as before. It is clear from the result below, that this is not a great design and clearly worse than the windowing method.





It is possible to improve the frequency sampling method in the following ways:

(a) Make the filter transitions smooth over $\Delta\omega$ width

(b) Oversample and do least squares fit **(Now can't use IDFT)**

(c) Use non-uniform points with more points near the transition **(Now can't use IDFT)**

4.3 IIR Digital Filter Design

Continuous Time Filters

Classical continuous-time filters optimise tradeoff: passband ripple v stopband ripple v transition width. This means there are explicit formulae for pole/zero positions.

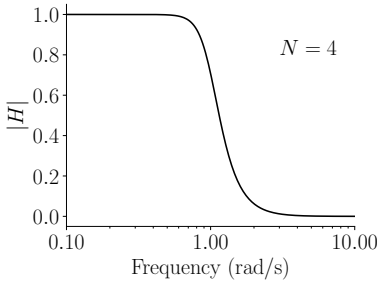
Butterworth Filter

$$G^2(\Omega) = |H(j\Omega)|^2 = \frac{1}{1+\Omega^{2N}}$$

(1) Monotonic $\forall\Omega$

(2) $G(\Omega) = 1 - \frac{1}{2}\Omega^{2N} + \frac{3}{8}\Omega^{4N} + \cdots$

Maximally flat: $2N-1$ derivatives are zero



Bilinear Transform

The bilinear transform is a widely used one-to-one invertible mapping. It involves a change variable:

$$z = \frac{\alpha + s}{\alpha - s} \Leftrightarrow s = \alpha\frac{z-1}{z+1}$$

(a) $\Re$ axis $(s) \leftrightarrow \Re$ axis (z)

(b) $\Im$ axis $(s) \leftrightarrow$ Unit circle (z)

Proof:

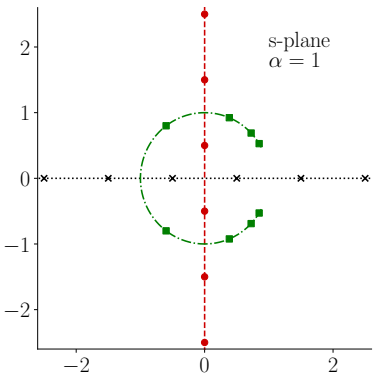
$$z = e^{j\omega} \Leftrightarrow s = \alpha\frac{e^{j\omega}-1}{e^{j\omega}+1} = \alpha\frac{e^{j\frac{\omega}{2}} - e^{-j\frac{\omega}{2}}}{e^{j\frac{\omega}{2}} + e^{-j\frac{\omega}{2}}} = j\omega\tan\frac{\omega}{2} = j\Omega$$

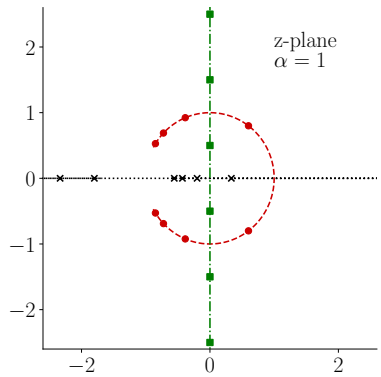
(c) Left half plane $(s) \leftrightarrow$ inside of unit circle (z)

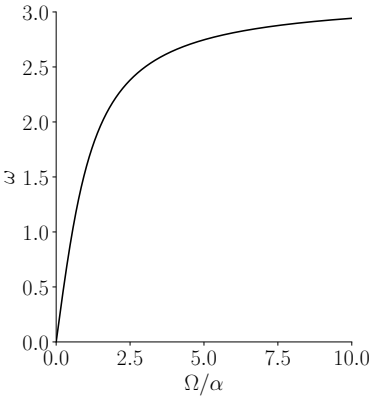
Proof:

$$s = x + jy \Leftrightarrow |z|^2 = \frac{|(\alpha+x)+jy|^2}{|(\alpha+x)-jy|^2} = \frac{\alpha^2+2\alpha x+x^2+y^2}{\alpha^2-2\alpha x+x^2+y^2} = 1 + \frac{4\alpha x}{(\alpha-x)^2+y^2}, \quad x < 0 \leftrightarrow |z| < 1$$

(d) Unit circle $(s) \leftrightarrow \Im$ axis (z)







Example:

$$H(s) = \frac{1}{s^2 + 0.2s + 4} \text{ and } \alpha = 1$$

Substitute: $s = \alpha\frac{z-1}{z+1}$

$$H(z) = \frac{1}{\left(\frac{z-1}{z+1}\right)^2 + 0.2\left(\frac{z-1}{z+1}\right) + 4} = \frac{(z+1)^2}{(z-1)^2 + 0.2(z-1)(z+1) + 4(z+1)^2}$$
$$= \frac{z^2 + 2z + 1}{5.2z^2 + 6z + 4.8} = 0.19\frac{1 + 2z^{-1} + z^{-2}}{1 + 1.15z^{-1} + 0.92z^{-2}}$$

Frequency response is identical in both magnitude and phase, however, the frequency axis has been distorted.

Frequency mapping: $\omega = 2\tan^{-1}\frac{\Omega}{\alpha}$

$$\Omega = [\alpha \quad 2\alpha \quad 3\alpha \quad 4\alpha \quad 5\alpha] \rightarrow \omega = [1.6 \quad 2.2 \quad 2.5 \quad 2.65 \quad 2.75]$$

Choosing α:

$$\text{Set } \alpha = \frac{\Omega_c}{\tan\frac{1}{2}\omega_c} \text{ to map } \Omega_c \rightarrow \omega_c$$

$$\text{Set } \alpha = 2f_s = \frac{2}{T} \text{ to map low frequencies to themselves.}$$

Impulse Invariance

Bilinear transform works well for a low-pass filter but the non-linear compression of the frequency distorts any other response. The impulse invariance transformation is an alternative method that obtains the discrete-time filter by sampling the impulse response of the continuous-time filter.

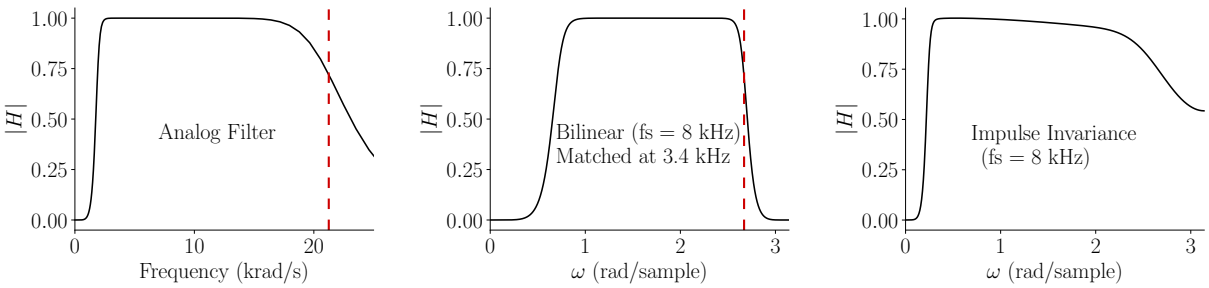
$$H(s) \xrightarrow{\mathscr{L}^{-1}} h(t) \xrightarrow{\text{sample}} h[n] = T \times h(nT) \xrightarrow{\mathscr{Z}} H(z)$$

Properties:

Advantages:	Impulse response is correct with no distortion of frequency axis.
Disadvantages:	The frequency response is aliased.

Example:

Standard telephone (300 to 3400 Hz bandpass) filter



4.4 Digital Filter Structures

Direct Forms

If an IIR filter has a transfer function:

$$H(z) = \frac{P(z)}{D(z)} = \frac{p[0] + p[1]z^{-1} + p[2]z^{-2} + \cdots + p[M-1]z^{-(M-1)} + p[M]z^{-M}}{1 + d[1]z^{-1} + d[2]z^{-2} + \cdots + d[N-1]z^{-(N-1)} + d[N]z^{-N}}$$

Then direct forms use coefficients $d[k]$ and $p[k]$ directly. This can be implemented as a cascade of two filter sections where:

$$H_1(z) = \frac{W(z)}{X(z)} = P(z) = p[0] + p[1]z^{-1} + p[2]z^{-2} + \cdots + p[M-1]z^{-(M-1)} + p[M]z^{-M}$$

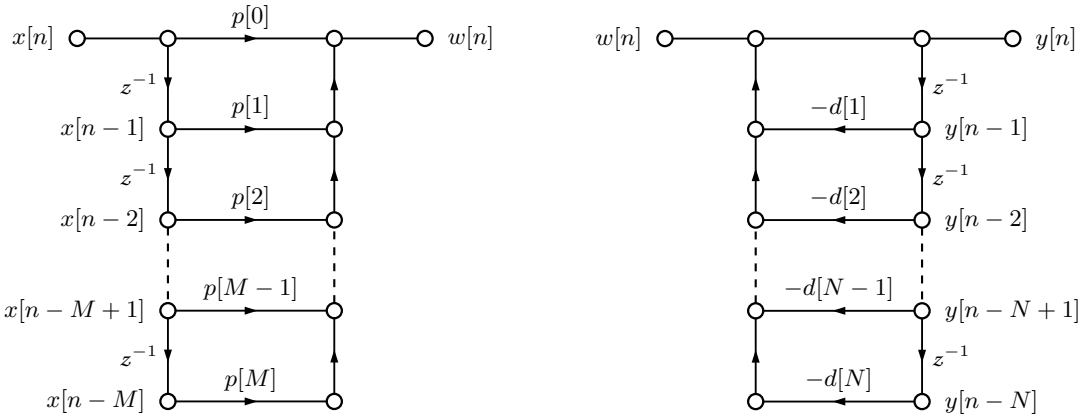
and

$$H_2(z) = \frac{Y(z)}{W(z)} = \frac{1}{D(z)} = \frac{1}{1 + d[1]z^{-1} + d[2]z^{-2} + \cdots + d[N-1]z^{-(N-1)} + d[N]z^{-N}}$$

Note that $H_1(z)$ can be seen as an FIR filter and the time-domain representation of $H_2(z)$ is given by:

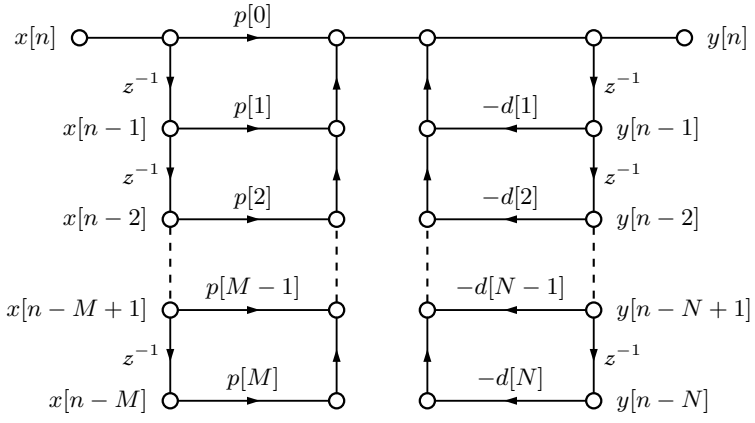
$$y[n] = w[n] - d[1]y[n-1] - d[2]y[n-2] - \cdots - d[N]y[n-N]$$

So the filter section $H_1(z)$ and $H_2(z)$ can be realised as shown :



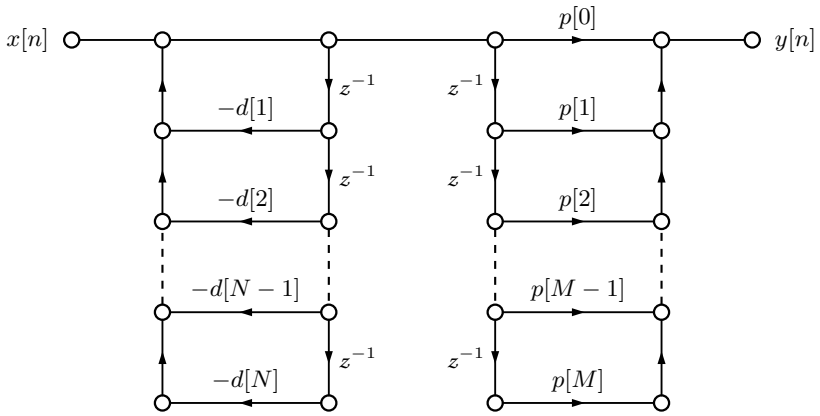
Direct Form I

Direct form I can be viewed as $P(z)$ followed by $\frac{1}{D(z)}$. This leads to the realisation of the original IIR transfer function:

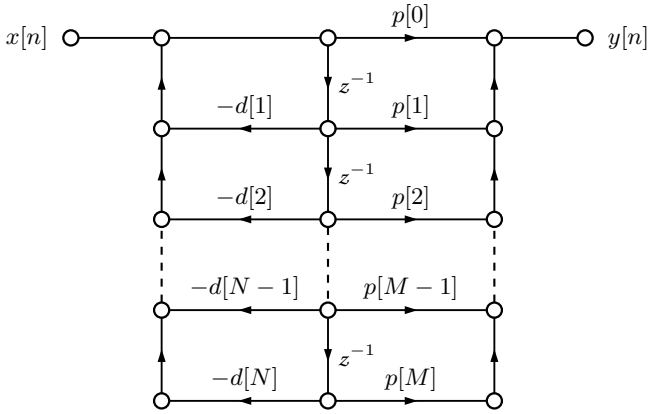


Direct Form II

Direct form II implements $\frac{1}{D(z)}$ followed by $P(z)$:



We observe that it is possible to share the delays which gives us the cononic structure below (also called direct form II):



The diagram above shows the case where $M = N$

So direct form II saves on delays (storage).

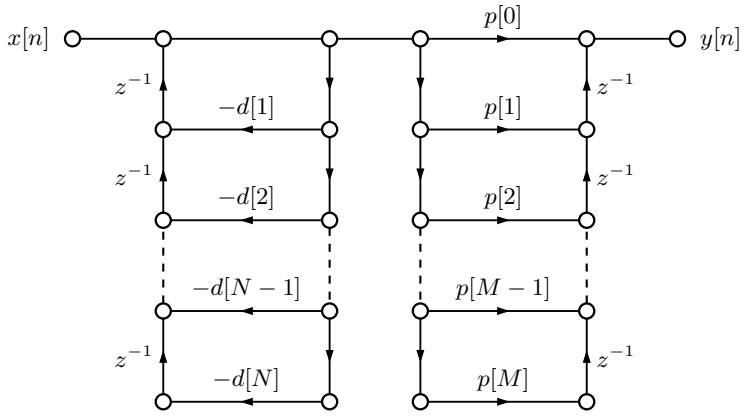
Transposed Forms

It is also possible to convert any structure into an equivalent transposed form. This is achieved in the following way:

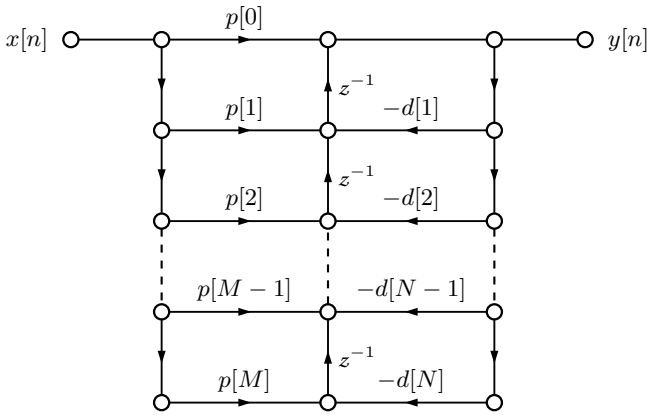
1. Reverse direction of each interconnection
2. Reverse direction of each multiplier
3. Change junctions to adders and vice-versa
4. Interchange the input and output signals

Check: A valid structure must never have any feedback loops that don't go through a delay (z^{-1} block).

Direct Form I Transposed



Direct Form II Transposed



The diagram above shows the case where $M = N$

Precision Issues

If all computations were exact, it would not make any difference which of the equivalent structures were used, however, this is never the case. There are two types of precision errors which are, coefficient precision and arithmetic precision.

- Coefficient precision:

  Coefficients can only be stored to finite precision and, therefore, are not exact. This means that the filter actually implemented is not correct. This is due to the fact that changes to the coefficients results in movment of the poles and zeros.

  The roots of high order polynomials can be very sensitive to small changes in coefficient values.

  A famous example being the Wilkinson's polynomial. In 1984, he described the personal impact of this discovery: *"Speaking for myself I regard it as the most traumatic experience in my career as a numerical analyst"*

- Arithmetic precision:

  It is also not possible to implement exact arithmetic calculations. These errors become particularly bad when calculating differences between two similar values:

  $$1.234567891.23456678 = 0.00000111 : 9 \text{ s.f.} \rightarrow 3 \text{ s.f.}$$

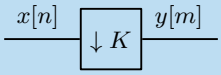
  Errors in arithmetic calculations have the effect of creating noise that is then filtered by the transfer function from the point of creation to the output.

5 Module 5 - Multirate Signal Processing

5.1 Multirate Systems

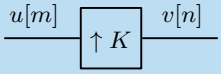
Building Blocks

Downsampling



$$y[m] = x[Km]$$

Upsampling

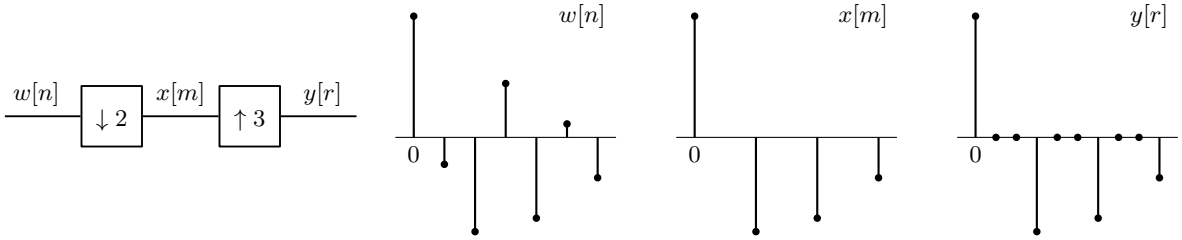


$$v[n] = \begin{cases} u\left[\frac{n}{K}\right], & n \mid K \\ 0, & \text{else} \end{cases}$$

Note: $a \mid b$ means that a divides into b exactly

Example

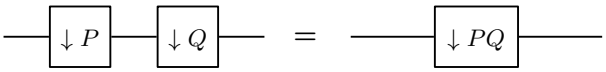
Downsample by 2 and then upsample by 3:



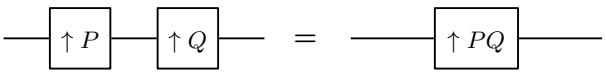
Notice the synchronisation, all signals have a sample at $n = 0$

Resampling Cascades

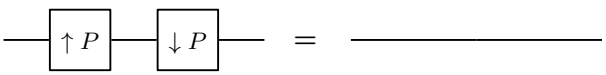
Successive downsamplers can be combined



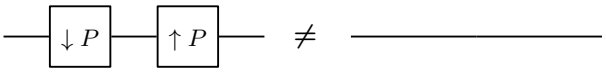
Successive upsamplers can be combined



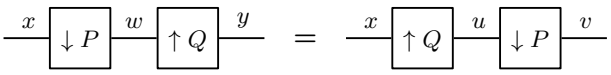
Upsampling can be inverted exactly



Downsampling permanently destroys information and thus is non invertible



If and, only if, P and Q are coprime then resampling can be interchanged



Proof:

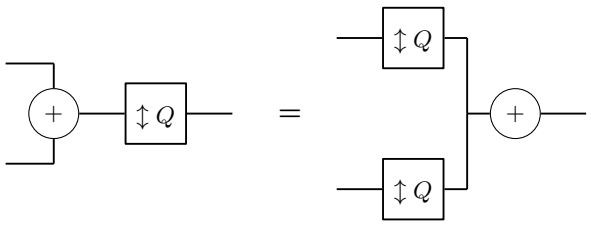
Left side: $y[n] = w\left[\frac{1}{Q}n\right] = x\left[\frac{P}{Q}n\right]$ if $Q \mid n$ else $y[n] = 0$

Right side: $v[n] = u[Pn] = x\left[\frac{P}{Q}n\right]$ if $Q \mid Pn$

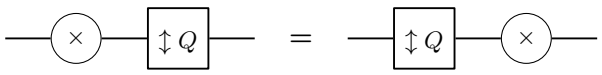
But $\{Q \mid Pn \Rightarrow Q|n\}$ iff P and Q are coprime

5.2 Noble Identities

Resamplers commute with addition

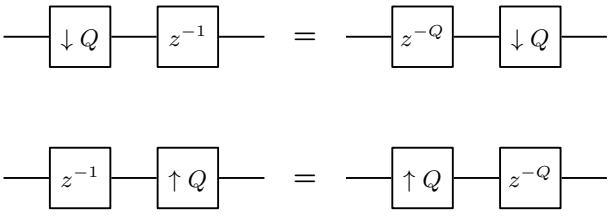


Resamplers commute with multiplication

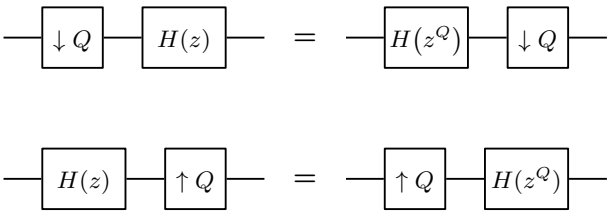


Note: $\updownarrow Q$ could be either $\uparrow Q$ or $\downarrow Q$

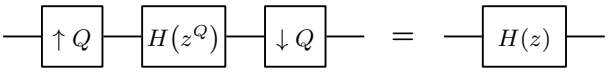
Delays must be multiplied by the resampling ratio



Noble identities: Exchange resamplers and filters



Corrollary:

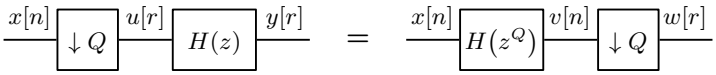


Example: $H(z) = h[0] + h[1]z^{-1} + h[2]z^{-2} + \cdots$ and $H(z^4) = h[0] + h[1]z^{-4} + h[2]z^{-8} + \cdots$

Noble identities proof

Downsampled Noble Identity

If $h_Q[n]$ is defined to be the impulse response of $H(z^Q)$.

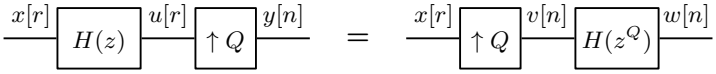


Let's assume that $h[r]$ is of length $M+1$ so that $h_Q[n]$ is of length $QM+1$. Note that $h_Q[n]=0$ except when $Q \mid n$ and that $h[r]=h_Q[Qr]$.

$$\begin{aligned} w[r] = v[Qr] &= \textstyle\sum_{s=0}^{QM} h_Q[s]x[Qr-s] \\ &= \textstyle\sum_{m=0}^{M} h_Q[Qm]x[Qr-Qm] = \sum_{m=0}^{M} h[m]x[Q(r-m)] \\ &= \textstyle\sum_{m=0}^{M} h[m]u[r-m] = y[r] \end{aligned}$$

Upsampled Noble Identity

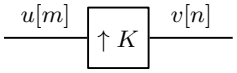
Note that $v[n]=0$ except when $Q \mid n$ and that $v[Qr]=x[r]$.

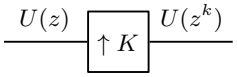


$$\begin{aligned} w[n] &= \textstyle\sum_{s=0}^{QM} h_Q[s]v[n-s] = \sum_{m=0}^{M} h_Q[Qm]v[n-Qm] \\ &= \textstyle\sum_{m=0}^{M} h[m]v[n-Qm] \\ &\text{if } Q \nmid n, \text{ then } v[n-Qm]=0 \;\forall\; m \text{ so } w[n]=0=y[n] \\ &\text{if } Q \mid n=Qr, \text{ then } w[Qr]=\textstyle\sum_{m=0}^{M} h[m]v[Qr-Qm]=\sum_{m=0}^{M} h[m]x[r-m]=u[r]=y[Qr] \end{aligned}$$

5.3 Upsampled z-transform

$V(z)=\sum_n v[n]z^{-n}=\sum_{n \text{ s.t. } K|n} u[\frac{n}{K}]z^{-n}=\sum_m u[m]z^{-Km}=U(z^K)$



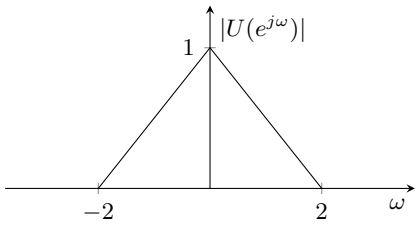


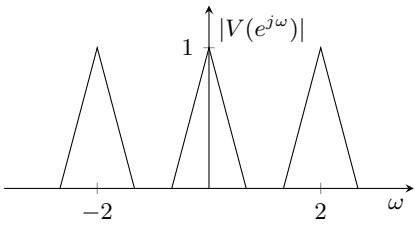
Spectrum: $V(e^{j\omega})=U(e^{jK\omega})$

The spectrum is horizontally shrunk and replicated K times.
Total energy unchanged and power (energy/sample) multiplied by $\frac{1}{k}$.
Upsampling normally followed by a low-pass filter to remove images.

Example:

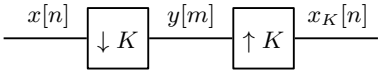
If $K=3$ then three images of the original spectrum are generated and the energy remains unchanged i.e. $\frac{1}{2\pi}\int |U(e^{j\omega})|^2 d\omega = \frac{1}{2\pi}\int |V(e^{j\omega})|^2 d\omega$.





5.4 Downsampled z-transform

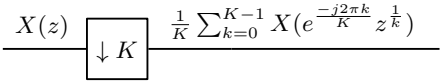
Define $c_K[n] = \delta_{K|n} = \frac{1}{K}\sum_{k=0}^{K-1} e^{\frac{j2\pi kn}{K}}$



Define $x_K[n] = \begin{cases} x[n], & K|n \\ 0, & K \nmid n \end{cases} = c_K[n]x[n]$

$$X_K(z) = \sum_n x_k[n]z^{-n} = \frac{1}{K}\sum_n \sum_{k=0}^{K-1} e^{\frac{j2\pi kn}{K}} x[n]z^{-n} = \frac{1}{K}\sum_{k=0}^{K-1}\sum_n e^{\frac{j2\pi kn}{K}} x[n]z^{-n}$$
$$= \frac{1}{K}\sum_{k=0}^{K-1}\sum_n x[n]\left(e^{\frac{-j2\pi k}{K}}z\right)^{-n} = \frac{1}{K}\sum_{k=0}^{K-1} X\left(e^{\frac{-j2\pi k}{K}}z\right)$$

$X_K(z) = Y(z^K) \Rightarrow Y(z) = X_K(z^{\frac{1}{K}}) = \frac{1}{k}\sum_{k=0}^{K-1} X(e^{\frac{-j2\pi k}{K}} z^{\frac{1}{k}})$



Spectrum:

$$Y(e^{j\omega}) = \frac{1}{k}\sum_{k=0}^{K-1} X(e^{\frac{j(\omega-2\pi k)}{K}}) = \frac{1}{k}\left(X(e^{\frac{j\omega}{K}}) + X(e^{\frac{j\omega}{K}-\frac{2\pi}{K}}) + X(e^{\frac{j\omega}{K}-\frac{4\pi}{K}}) + \cdots\right)$$

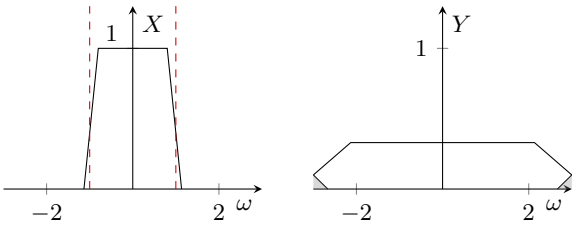
Average of K aliased versions, each expanded in ω by a factor of K. This is why downsampling is normally preceded by a low-pass filter to prevent aliasing.

Example 1:

$K = 3$

Not quite limited to $\pm\frac{\pi}{K}$. The shaded region highlights the resulting aliasing.

Energy decreases:
$\frac{1}{2\pi}\int |Y(e^{j\omega})|^2 d\omega \approx \frac{1}{K} \times \frac{1}{2\pi}\int |X(e^{j\omega})|^2 d\omega$

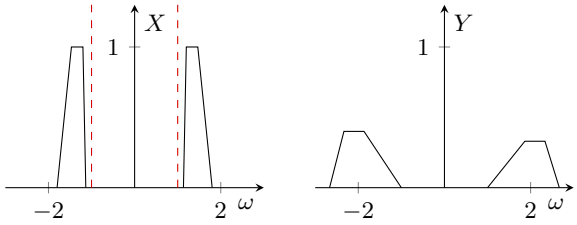


Example 2:

$K = 3$

All engergy between $\frac{\pi}{K} \le |\omega| < 2\frac{\pi}{K}$.

Therefore no aliasing.



No aliasing

There is no aliasing if all the energy is between $r\frac{\pi}{K} \le |\omega| < (r+1)\frac{\pi}{K}$ for some integer r.

The normal case being when $r = 0$ where all the energy is between $\frac{\pi}{K} \le |\omega| < \frac{\pi}{K}$.

Effects of downsampling

1. Total energy multiplied by $\approx \frac{1}{K} (= \frac{1}{K}$ if no aliasing).
2. The average power (energy/sample) $\approx$ is unchanged.

5.5 Perfect Reconstruction

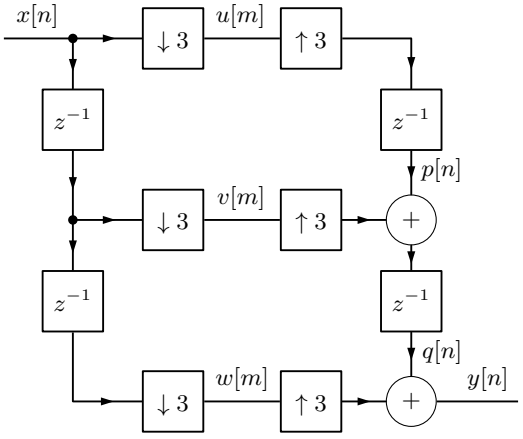
It is possible to achieve perfect reconstruction after downsampling, for example, if a given input sequence $x[n]$ is split into three streams at $\frac{1}{3}$ of the sample rate:

$$u[m] = x[3m], v[m] = x[3m-1], w[m] = x[3m-2]$$

And then if all the following sequences are upsampled and aligned by delays then, they can be added together to give:

$$y[n] = x[n-2]$$

$x[n]$	c	d	e	f	g	h	i	j	k	l	m	n
$u[m]$	c			f			i			l		
$p[n]$	-	c	-	-	f	-	-	i	-	-	l	
$v[m]$	b			e			h			k		
$q[n]$	-	b	c	-	e	f	-	h	i	-	k	l
$w[m]$	a			d			g			j		
$y[n]$	a	b	c	d	e	f	g	h	i	j	k	l

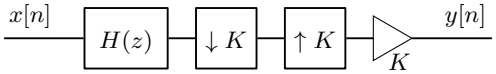


Therefore the output $y[n]$ in this case is just a delayed replica of the input $x[n]$.

5.6 Polyphase Filters

Maximum Frequency Decimation

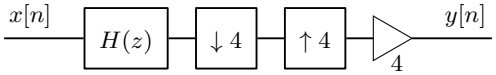
If the bandpass of a filter only occupies a small fraction of $[0, \pi]$, it is possible to downsample then upsample without losing any information.

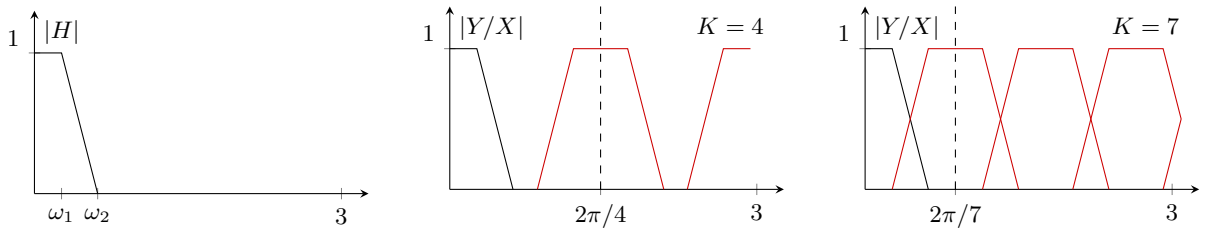


To avoid aliasing in the passband:

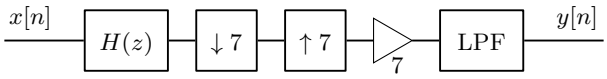
$$\frac{2\pi}{K} - \omega_2 \geq \quad \Rightarrow \quad K \leq \frac{2\pi}{\omega_1 - \omega_2}$$

Centre of the transition band must be $\leq$ intermediate Nyquist freq, $\frac{\pi}{K}$





The images spaced at $\frac{2\pi}{K}$ can be removed using another low-pass filter.

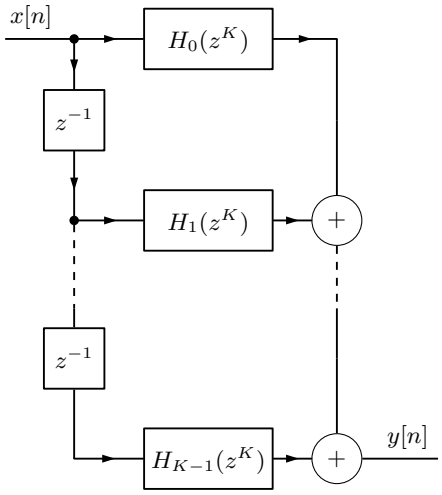


Note: the passband noise is equal to the noise floor at output of $H(z)$ plus $10\log_{10}(K-1)$ dB.

Polyphase Decomposition

The general case of a K-branch polyphase decomposition of the transfer function $H(z)$ of order N is of the form:

$$H(z) = \sum_{m=0}^{K-1} z^{-m} H_m(z^K)$$

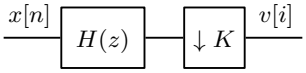


Proof:

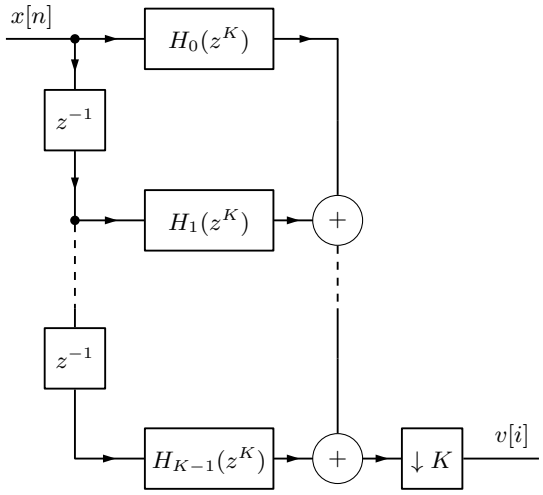
$$\begin{aligned}
H(z) &= \sum_{m=0}^{M} h[m] z^{-m} \\
&= \sum_{m=0}^{K-1} h[m] z^{-m} + \sum_{m=0}^{K-1} h[m+K] z^{-(m+K)} + \cdots \\
&= \sum_{r=0}^{R-1} \sum_{m=0}^{K-1} h[m+Kr] z^{-(m+Kr)} \\
&= \sum_{m=0}^{K-1} z^{-m} \sum_{r=0}^{R-1} h_m[r] z^{-Kr} \\
&\qquad\qquad \text{where } h_m[r] = h[m+Kr] \\
&= \sum_{m=0}^{K-1} z^{-m} H_m(z^K)
\end{aligned}$$

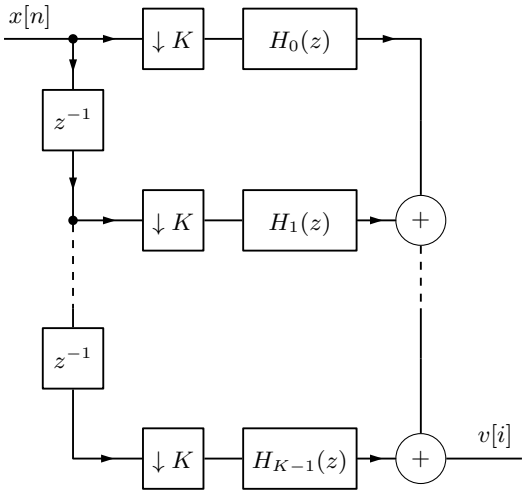
Polyphase Downsampler

If $H(z)$ is low-pass so that it is possible to downsample its output by K without the problem of aliasing.



By decomposing $H(z)$ into a a polyphase representation, it is possible to take advantage of the Noble identities. It is therefore possible to move the downsampling back through the adders and filters. Thus $H_m(z^K)$ turns into $H_m(z)$ at the lower sample rate. This has the effect of massively reducing the computation.

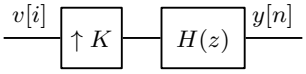




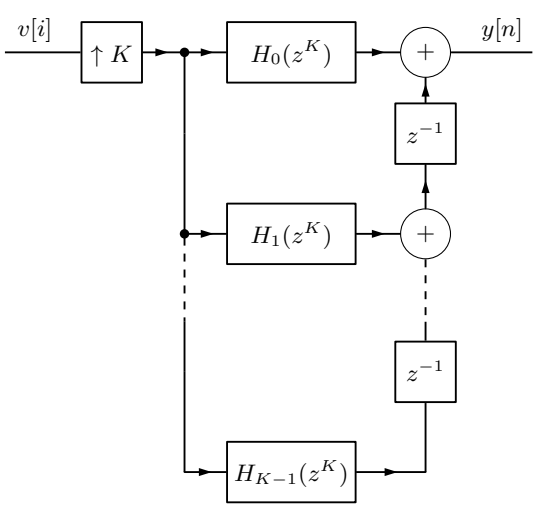
The sample rate at $v[i]$ is now lower, however, it is possible to restore the original sample rate by upsampling.

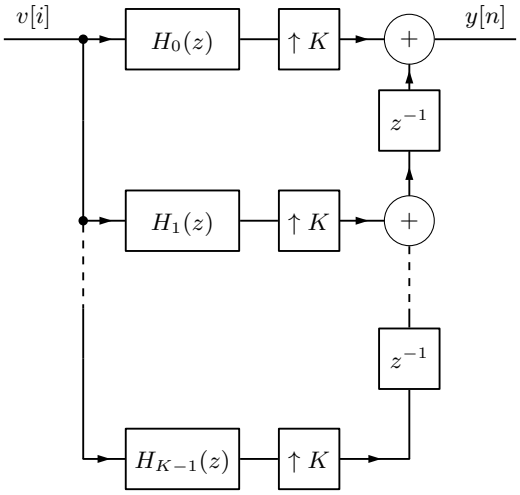
Polyphase Upsampler

Upsampling is always followed by a low-pass filter to remove the images



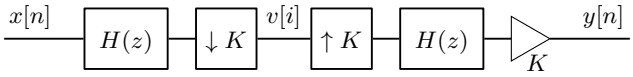
It is possible to use the same low-pass filter, $H(z)$, in polyphase form. In this case the delay z^{-m} after the filters. By taking advantage of the Noble identities, it is possible to move the upsampling forward through the filters. Thus $H_m(z^K)$ turns into $H_m(z)$ at the lower sample rate. This again has the effect of massively reducing the computation.





Complete Filter

The overall system implements:



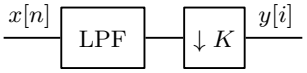
The extra gain of K is needed to compensate for the downsampling energy loss.

5.7 Resampling

The conditions required to change the sampling rate while preserving information:

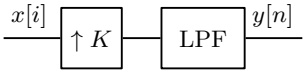
Downsample:

LPF to new Nyquist bandwidth: $\omega_c = \frac{\pi}{K}$



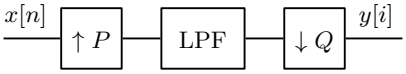
Upsample:

LPF to old Nyquist bandwidth: $\omega_c = \frac{\pi}{K}$



Rational ratio: $f_s \times \frac{P}{Q}$

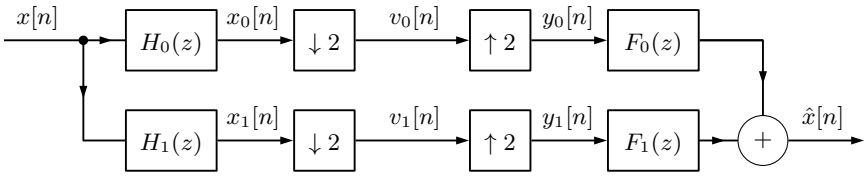
LPF to lower of old and new Nyquist bandwidth: $\omega_c = \frac{\pi}{\max(P,Q)}$



(a) Polyphase decomposition reduces computation by $K = \max(Q, P)$

(b) The transition band centre should be at the Nyquist frequency $\omega_c = \frac{\pi}{K}$

5.8 2-band Filterbank



$$X_m(z) = H_m(z)X(z), \quad \text{where } m \in \{0,1\}$$

$$V_m(z) = \frac{1}{K}\sum_{k=0}^{K-1} X_m(e^{\frac{-j2\pi k}{K}} z^{\frac{1}{k}}) = \frac{1}{2}\Big\{X_m(z^{\frac{1}{2}}) + X_m(-z^{\frac{1}{2}})\Big\}$$

$$Y_m(z) = V_m(z^2) = \frac{1}{2}\{X_m(z) + X_m(-z)\} = \frac{1}{2}\{H_m(z)X_m(z) + H_m(-z)X_m(-z)\}, \quad \text{where } k = 2$$

$$\begin{aligned}
\hat{X}(z) &= \begin{bmatrix} Y_0(z) & Y_1(-z) \end{bmatrix} \begin{bmatrix} F_0(z) \\ F_1(z) \end{bmatrix} \\
&= \frac{1}{2}\begin{bmatrix} X(z) & X(-z) \end{bmatrix} \begin{bmatrix} H_0(z) & H_1(z) \\ H_0(-z) & H_1(-z) \end{bmatrix} \begin{bmatrix} F_0(z) \\ F_1(z) \end{bmatrix} \\
&= \begin{bmatrix} X(z) & X(-z) \end{bmatrix} \begin{bmatrix} T(z) \\ A(z) \end{bmatrix}, \quad \text{where } X(-z)A(z) \text{ is the 'aliased' term.}
\end{aligned}$$

we want:

$$T(z) = \frac{1}{2}\{H_0(z)F_0(z) + H_1(z)F_1(z)\} = z^{-d}$$

$$A(z) = \frac{1}{2}\{H_0(-z)F_0(z) + H_1(-z)F_1(z)\} = 0$$

For perfect reconstruction without aliasing, we require:

$$\frac{1}{2}\begin{bmatrix} H_0(z) & H_1(z) \\ H_0(-z) & H_1(-z) \end{bmatrix} \begin{bmatrix} F_0(z) \\ F_1(z) \end{bmatrix} = \begin{bmatrix} z^{-d} \\ 0 \end{bmatrix}$$

Hence:

$$\begin{aligned}
\begin{bmatrix} F_0(z) \\ F_1(z) \end{bmatrix} &= \begin{bmatrix} H_0(z) & H_1(z) \\ H_0(-z) & H_1(-z) \end{bmatrix}^{-1} \begin{bmatrix} 2z^{-d} \\ 0 \end{bmatrix} \\
&= \frac{2z^{-d}}{H_0(z)H_1(-z) - H_1(z)H_0(-z)} \begin{bmatrix} H_1(-z) & -H_1(z) \\ -H_0(-z) & H_0(z) \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
&= \frac{2z^{-d}}{H_0(z)H_1(-z) - H_1(z)H_0(-z)} \begin{bmatrix} H_1(-z) \\ -H_0(-z) \end{bmatrix}
\end{aligned}$$

For all filters to be FIR, we need the denominator to be

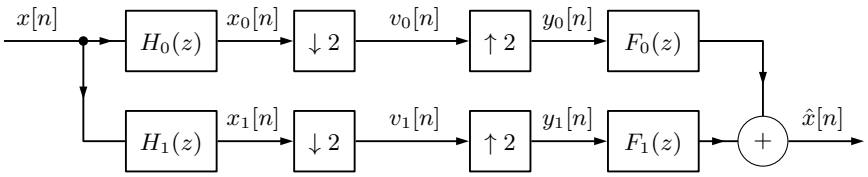
$$H_0(z)H_1(-z) - H_1(z)H_0(-z) = cz^{-k} \quad \text{(scaling factor and delay)}$$

which implies:

$$\begin{bmatrix} F_0(z) \\ F_1(z) \end{bmatrix} = \frac{2}{c}z^{k-d} \begin{bmatrix} H_1(-z) \\ -H_0(-z) \end{bmatrix} \overset{d=k}{=} \frac{2}{c} \begin{bmatrix} H_1(-z) \\ -H_0(-z) \end{bmatrix}$$

$H_i(z)$ is scaled by $c^{\frac{1}{2}}$ and $F_i(z)$ is scaled by $c^{-\frac{1}{2}}$

5.9 Quadrature Mirror Filterbank (QMF)



QMF satisfies:

(a) $H_0(z)$ is real and causal

(b) $H_1(z) = H_0(-z)$: i.e. $|H_0(e^{j\omega})|$ is reflected around $\omega = \frac{\pi}{2}$

(c) $F_0(z) = H_1(-z) = H_0(z)$

(d) $F_1(z) = -H_0(-z) = -H_1(z)$

QMF is alias-free:

$$\begin{aligned} A(z) &= \frac{1}{2}\{H_0(-z)F_0(z) + H_1(-z)F_1(z)\} \\ &= \frac{1}{2}\{H_1(z)H_0(z) - H_0(z)H_1(z)\} = 0 \end{aligned}$$

QMF Transfer Function:

$$\begin{aligned} T(z) &= H_0(z)F_0(z) + H_1(z)F_1(z) \\ &= H_0^2(z) - H_1^2(z) = H_0^2(z) - H_0^2(-z) \end{aligned}$$

5.10 Polyphase QMF

Polyphase decomposition:

$$\begin{aligned} H_0(z) &= E_0(z^2) + z^{-1}E_1(z^2) \\ H_1(z) &= H_0(-z) = E_0(z^2) - z^{-1}E_1(z^2) \\ F_0(z) &= H_0(z) = E_0(z^2) + z^{-1}E_1(z^2) \\ F_1(z) &= -H_0(-z) = -E_0(z^2) + z^{-1}E_1(z^2) \end{aligned}$$

The matrix form of the analysis side:

$$\begin{bmatrix} H_0(z) \\ H_1(z) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} E_0(z^2) \\ z^{-1}E_1(z^2) \end{bmatrix}$$

Can be written in this way:

$$\begin{bmatrix} H_0(z) & H_1(z) \end{bmatrix} = \begin{bmatrix} E_0(z^2) & z^{-1}E_1(z^2) \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

The matrix form of the synthesis side:

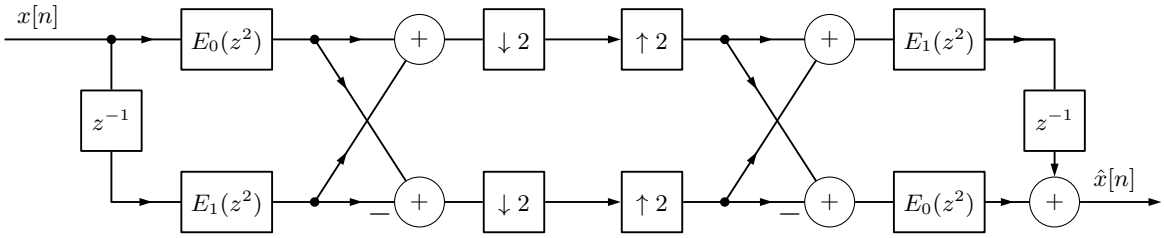
$$\begin{bmatrix} F_0(z) \\ F_1(z) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} z^{-1}E_1(z^2) \\ E_0(z^2) \end{bmatrix}$$

Therefore

$$\hat{X}(z) = X(z) \begin{bmatrix} H_0(z) & H_1(z) \end{bmatrix} \begin{bmatrix} F_0(z) \\ F_1(z) \end{bmatrix}$$

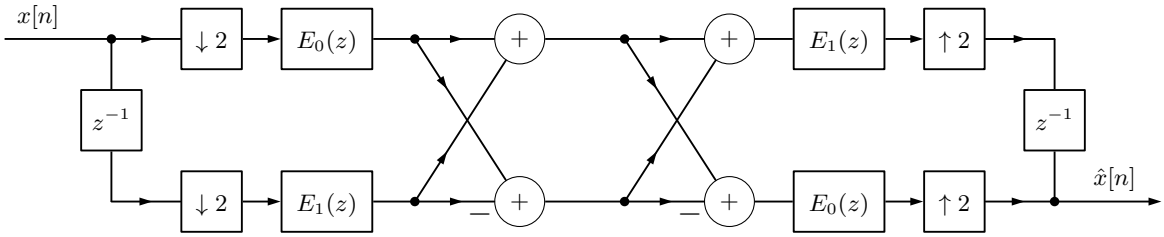
By substitution can be written in this way:

$$\hat{X}(z) = X(z) \begin{bmatrix} E_0(z^2) & z^{-1}E_1(z^2) \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} E_0(z^2) \\ z^{-1}E_1(z^2) \end{bmatrix}$$



This reduces the computational complexity by a factor of 2 as the processing is now performed on half the samples.

By applying the Noble identities, it is possible to obtain the following structure in which all filtering is carried out efficiently at the lowest possible sampling rate



This reduces the computational complexity again by a factor of 2 and, therefore, the polyphase QMF reduces the overall computational complexity by a factor of 4.

References

[1] Patrick A. Naylor, *lecture slides*. Course: EE3-07 Digital Signal Processing.
`Imperial College London`

[2] Mike Brookes, *lecture slides*. Course: EE4-12 Digital Signal Processing and Digital Filters.
`Imperial College London`

[3] Roy S. C. Dutta, *lecture slides* Course: Digital Signal Processing.
`Indian Institute of Technology`

[4] John G. Proakis and Dimitris G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications, 1996.*
`Prentice-Hall, Inc`

[5] Sanjit K. Mitra, *Digital Signal Processing: A Computer-Based Approach, 2001.*
`McGraw-Hill School Education Group`

[6] Strang, Gilbert, *Computational Science and Engineering, 2007.*
`Wellesley, MA: Wellesley-Cambridge Press`

[7] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding, 1995.*
`Prentice-Hall, Inc`