(Digital) Signal Processing — Overview

- In the first part, you have studied a good deal of background theory;
- we want to explore some more (particularly digital) signal processing theory, applications, and architectures;
- digital signal processing (DSP) is based on theory that goes back more than 200 years to Norbert Gauss and others;
- DSP has seen a revolution due to the arrival of inexpensive computational devices in the 1990;
- DSP has penetrated almost every aspect of life: Texas Instruments claims that every 10 minutes you use one of their dedicated digital signal processors.

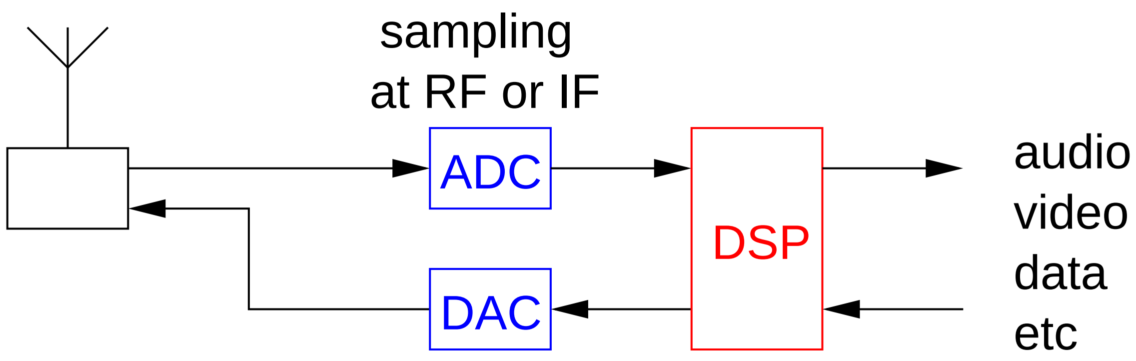
DSP Applications

- Multimedia: compression and coding algorithms for audio, video, and still images; acoustic echo control for teleconferencing;
- Audio: audio surround sound, MP3, seamless audio interfaces;
- Communications: equalisation, detection, smart antenna processing;
- Biomedical: signal detection; source separation for ECG or EEG; digital hearing aids;
- Defense: radar/sonar; detection and guidance systems;
- Automotive: GPS navigation; engine management;

DSP Perspective: Software Defined Radio

- Reliability, repeatability and programmability of DSP has widely replaced analogue systems in consumer and industrial markets;

- an ambitious future system relying on DSP is a *software defined radio*:

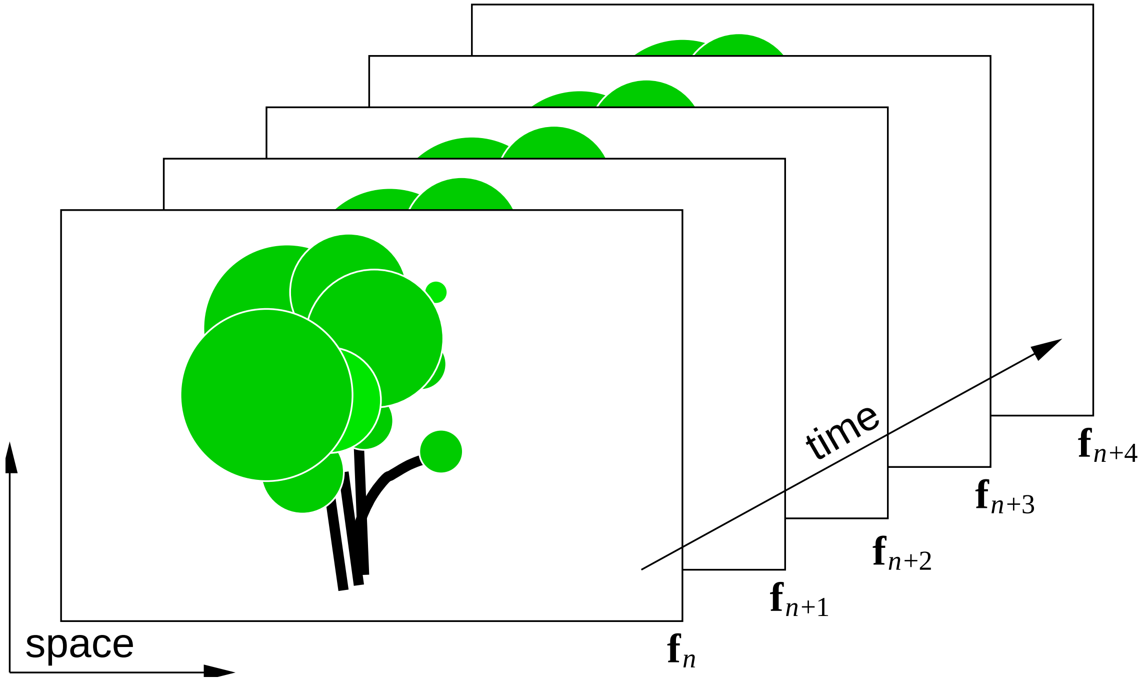


Sampling at an intermediate or even radio frequency (IF or RF) permits flexibility of the modulation stage and use of advanced DSP techniques for synchronisation, equalisation, carrier offset recovery etc.

Notation — Time Domain

- Continuous time signal: $x(t)$ with $t \in \mathbb{R}$; example: analogue speech signal or any other voltage signal;

- discrete time signal $x[n]$ with $n \in \mathbb{Z}$; example: daily stock price, video signal:



- vector quantity: $\mathbf{x}$, if time dependent $\mathbf{x}[n]$ or $\mathbf{x}_n$

- matrix quantity: $\mathbf{T}$

Notation — Transform Domain

- Fourier transform:

$$x(t) \circ\!\!-\!\!\bullet\; X(j\omega) \qquad X(j\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t}\, dt \tag{1}$$

$$x[n] \circ\!\!-\!\!\bullet\; X(e^{j\Omega}) \qquad X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\Omega n} \tag{2}$$

with angular frequency $\omega = 2\pi f$ and normalised angular frequency $\Omega = 2\pi\omega/\omega_\mathrm{s}$ for sampling at ω_s.

- note that $X(e^{j\Omega})$ is periodic with 2π.

- z-transform: $H(z) \bullet\!\!-\!\!\circ\; h[n]$ such that $H(z) = \sum_{n=-\infty}^{\infty} h[n]\, z^{-n}$

Notation — Stochastic Signals

- A stochastic signal $x[n]$ can be characterised by its probability density function (PDF);

- often we are interested in its autocorrelation function

$$r_{xx}[\tau] = \mathcal{E}\{x[n]x^*[n-\tau]\} \tag{3}$$

  with lag τ, complex conjugation operator $\{\cdot\}^*$ and expectation operator $\mathcal{E}\{\cdot\}$;

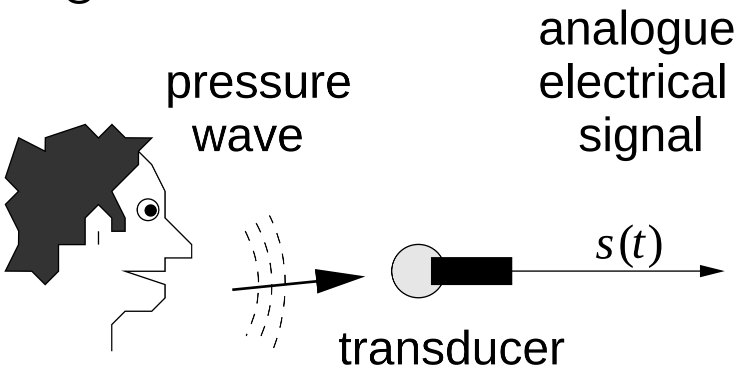
- power of $x[n]$ is given by $r_{xx}[0]$; if $x[n]$ is zero mean, then the variance $\sigma^2_{xx} = r_{xx}[0]$;

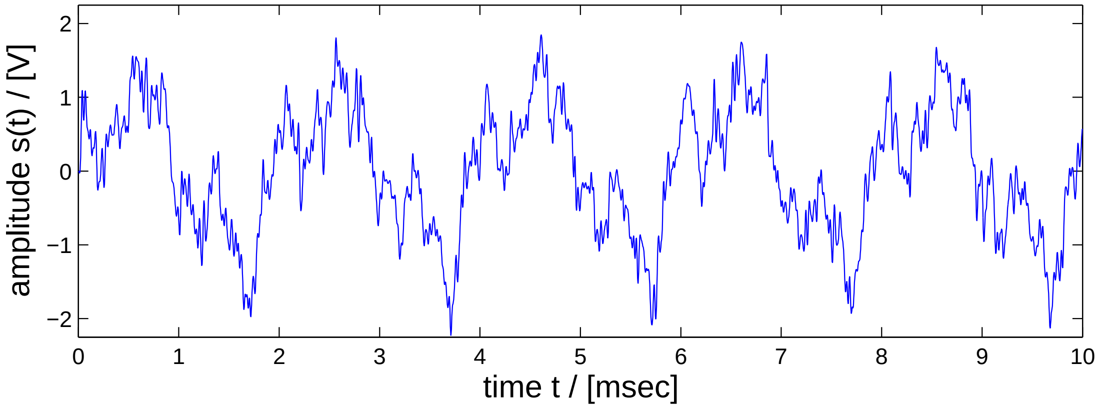
- power spectral density (PSD) of $x[n]$:

$$R_{xx}(e^{j\Omega}) \bullet\!\!-\!\!\circ\; r_{xx}[\tau] \qquad \text{such that} \quad R_{xx}(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} r_{xx}[\tau] e^{-j\Omega\tau} \tag{4}$$

Electrical (Voltage) Signals

- A signal is a variation over time of a measurable quanity that conveys information of some form;
- a sensor (actuator) can convert a real world signal to (from) a convenient voltage signal:



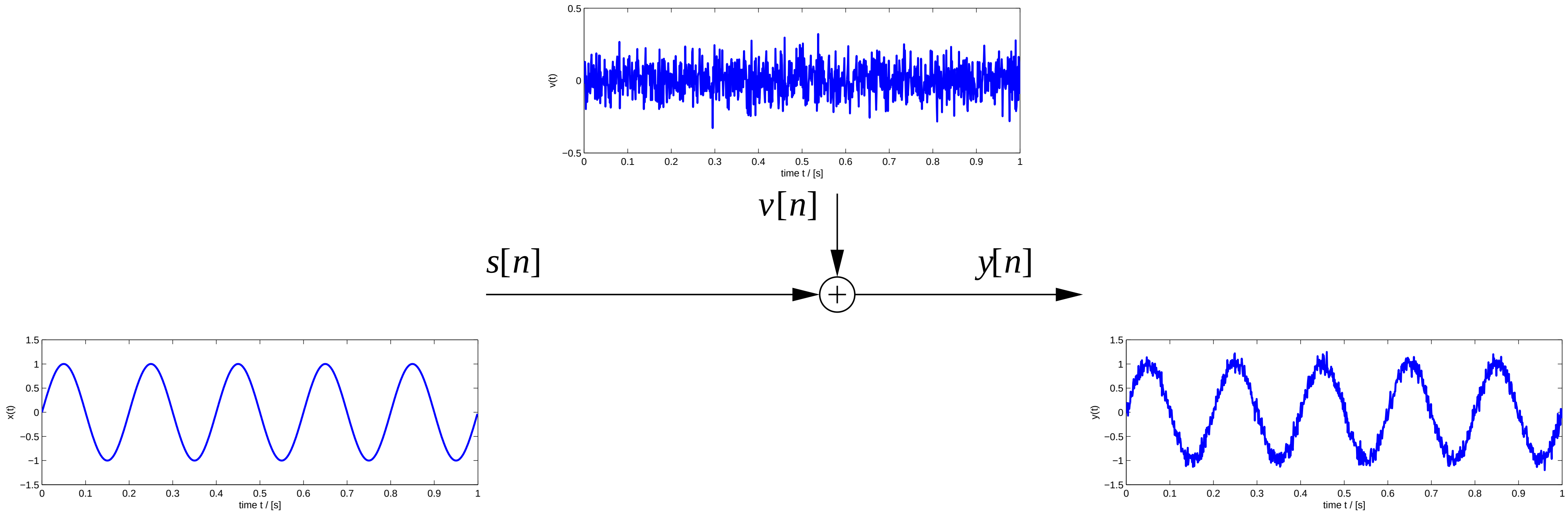


- other conversion examples: pressure — piezoelement; light — photosensor; vibration — accelerometer; electro-magnetic — antenna.

Amplification and Noise

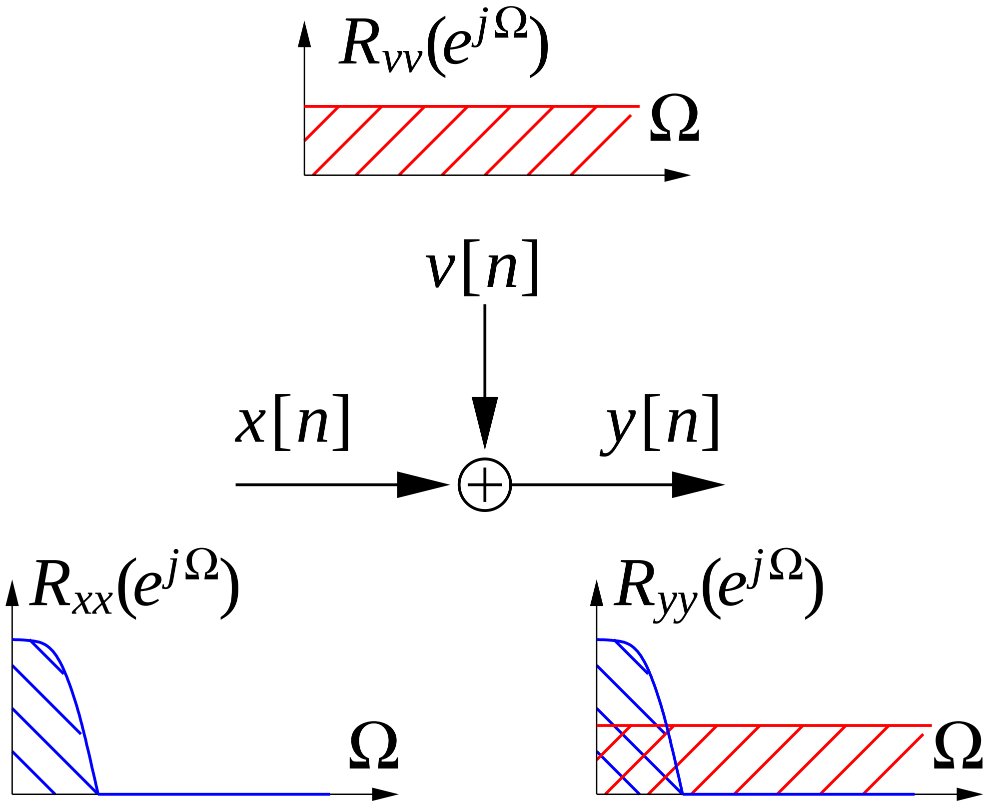
- Many real world signals yield small amplitudes (e.g. EEG order of $10^{-6}v$), and amplification is required prior to further processing;

- generally such signals are corrupted by noise; we assume an additive noise model:



Additive Noise in the Frequency Domain

- Additive noise model in the frequency domain:



- $x[n]$ is the signal, $v[n]$ the corrupting noise;
- we assume $x[n]$ and $v[n]$ to be zero mean processes and mutually uncorrelated;
- measure signal quality: signal-to-noise ratio (SNR);
- signal and noise power can be determined from the PSDs.

- from assumptions $\longrightarrow$ $\sigma^2_{yy} = \sigma^2_{xx} + \sigma^2_{vv}$ and $P_{yy}(e^{j\Omega}) = P_{xx}(e^{j\Omega}) + P_{vv}(e^{j\Omega})$.

Signal-to-Noise Ratio

- The signal to noise ratio is a power ratio:

$$\text{SNR} = \frac{\text{signal power}}{\text{noise power}} \tag{5}$$

- for zero-mean signals: $\text{SNR} = \sigma^2_{\text{signal}}/\sigma^2_{\text{noise}}$;

- the range of values to be measured may span several orders of magnitude (such as the human hearing); therefore a logarithmic scale has been introduced:

$$\text{SNR}_{\text{dB}} = 10 \cdot \log_{10} \frac{\sigma^2_{\text{signal}}}{\sigma^2_{\text{noise}}} = 20 \cdot \log_{10} \frac{\sigma_{\text{signal}}}{\sigma_{\text{noise}}} \quad \text{[deciBel, dB]} \tag{6}$$

- Examples: $\sigma^2_{\text{signal}} = 1,000 \cdot \sigma^2_{\text{noise}} \longrightarrow \text{SNR}_{\text{dB}} = 30$ dB;
  $\sigma^2_{\text{signal}} = 1,000,000 \cdot \sigma^2_{\text{noise}} \longrightarrow \text{SNR}_{\text{dB}} = 60$ dB.

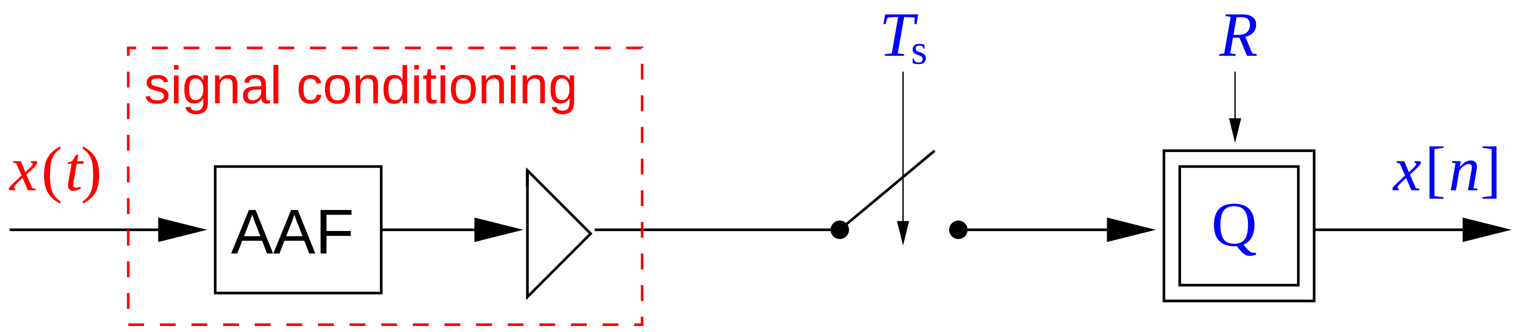
Analogue-to-Digital and Digital-to-Analogue Conversion

- In order to interface with the real world, we will need to convert between analogue and digital signals;
- this is accomplished by analogue-to-digital and digital-to-analogue converters (ADCs / DACs);
- analogue signals, such as a speech signal, may require a conversion into the digital domain for applying sophisticated source coding algorithms:

- digital signals need to be converter to analogue quantities for transmission over an antenna link:

Analogue-to-Digital Conversion

- An ADC requires conditioning of the analogue signal, sampling, and quantisation:



- sampling: the selection of the sampling period T_s will affect the possible representation of the frequency content of $x(t)$ by $x[n]$;

- quantisation: the word length R will influence resolution, or — as derived in slide 17 — the signal-to-quantisation noise ratio (SQNR);

- signal conditioning: contains an anti-alias filter (AAF) and a gain in order to adjust the amplitude of $x(t)$ to the input range of the quantiser Q.

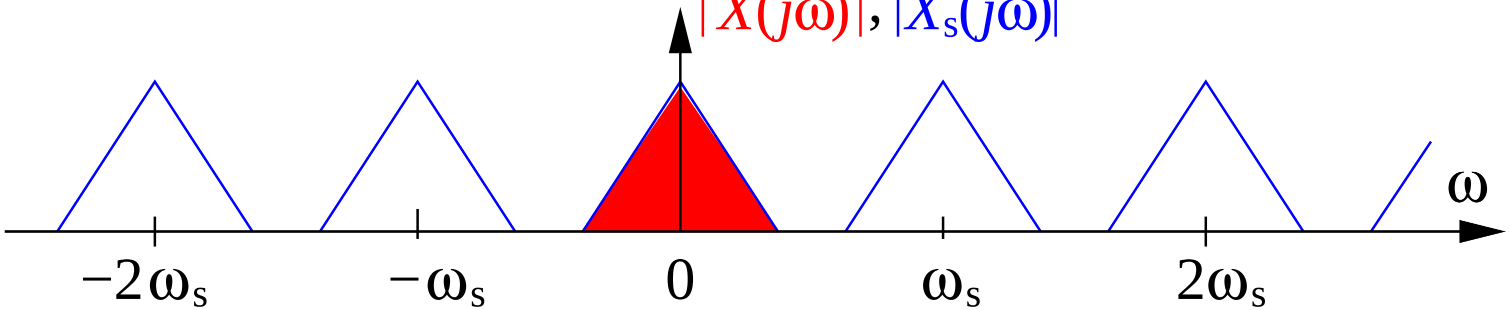
Sampling

- Sampling $x[n]$ from a continuous $x(t)$: $x_{\rm s}(t) = x(t) \cdot \sum_{n=-\infty}^{\infty} \delta(t - nT_{\rm s})$;

- remembering the Fourier series theorem and the duality property of the Fourier transform, it is clear that the frequency domain of the sampled signal $x_{\rm s}(t)$ has been periodised, w.r.t. $\omega_{\rm s} = 2\pi/T_{\rm s}$:

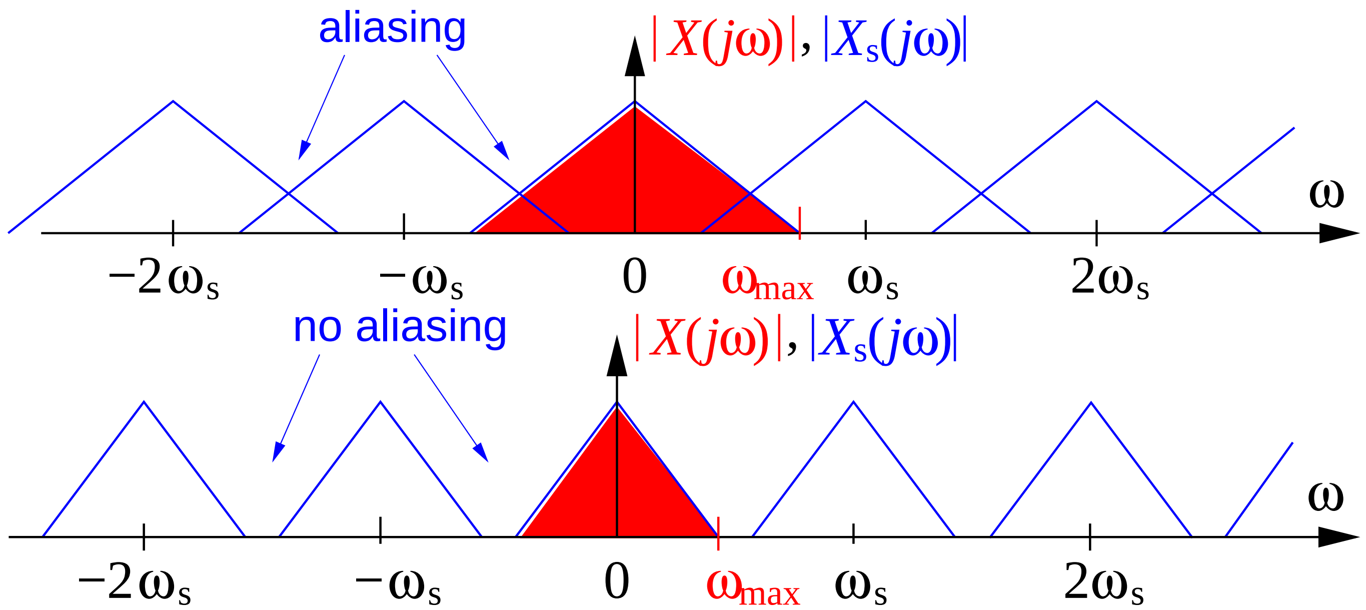
$$X_{\rm s}(j\omega) = \sum_{n=-\infty}^{\infty} X(j(\omega - n\omega_{\rm s})) \qquad (7)$$

- hence



Nyquist Rate and Aliasing

- Nyquist sampling theorem: all information of a continuous time signal is retained if spectral repetitions do not overlap; theoretically, the analogue signal can be perfectly reconstructed from the sampled version;

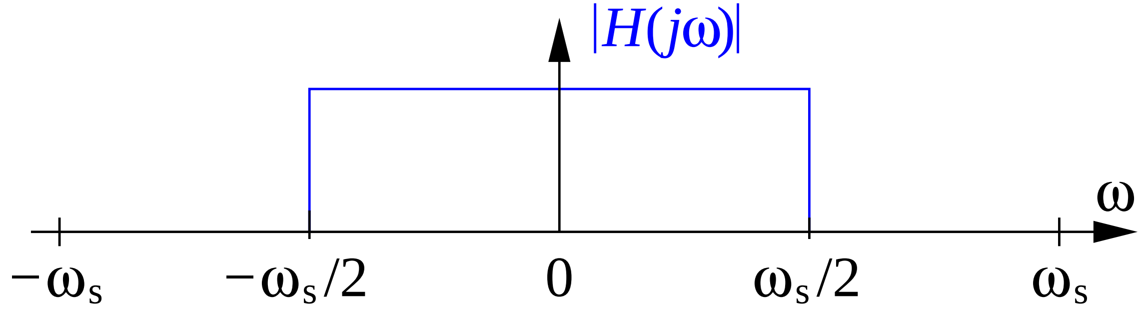


- hence the requirement:

$$\omega_s \quad > \quad \omega_N \quad = \quad 2 \cdot \omega_{max} \qquad \text{with Nyqust rate } \omega_N \tag{8}$$

Signal Conditioning

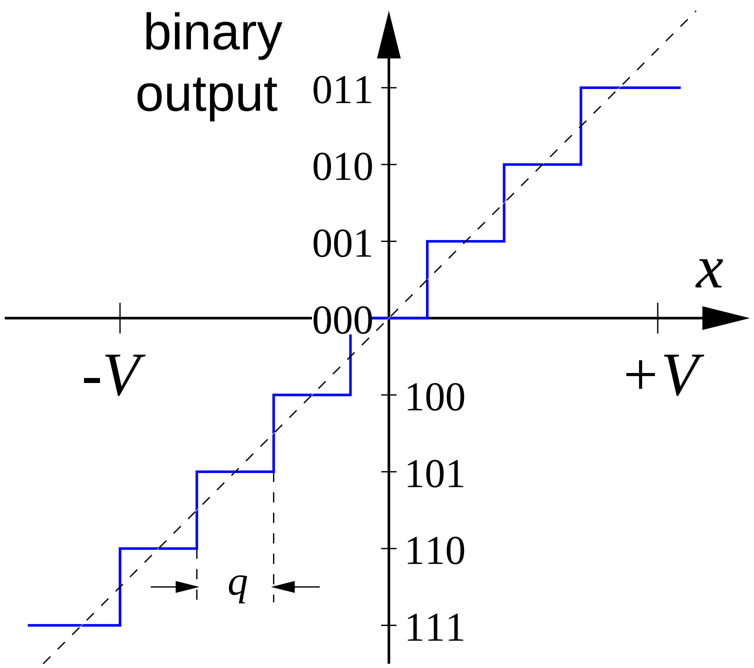
- For a perfect signal representation, we have to sample a signal *at least at the Nyquist rate*;
- in order to avoid aliasing, we must fulfil: $\omega_{\mathrm{max}} < \omega_{\mathrm{s}}/2$;
- if the signal's content is not exactly known, we can guarantee this by conditioning the input signal to the ADC by an anti-alias filter:



- this filter removes all frequency components above $\omega_{\mathrm{s}}/2$, which would otherwise alias in the sampling stage.

Quantisation

- An R-bit quantiser approximates a straight line with 2^R levels; example with $R = 3$ and asymmetric characteristic $\longrightarrow$



- the input signal has to use the range $[-V; +V]$ as best as possible, otherwise clippling or coarse resolution result;

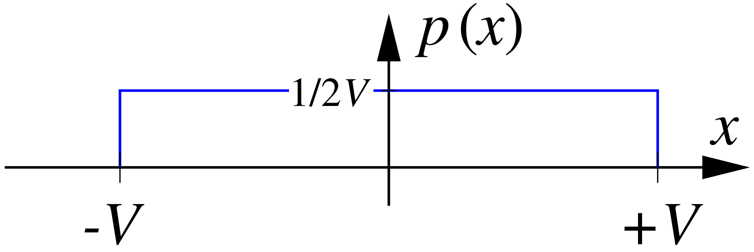
- quantisation noise: maximum $q/2$, uniformly distributed, additive;

- quantisation noise power (no clipping!)

$$\sigma_e^2 = \int_{-q/2}^{q/2} e^2 \cdot p(e) de = \frac{1}{q} \left[\frac{1}{3} e^3 \right]_{-q/2}^{q/2} = \frac{q^2}{12} \qquad (9)$$

Signal-to-Quantisation Noise Ratio

- Consider a uniformly distributed input signal $x(t)$:



- therefore, the signal power is

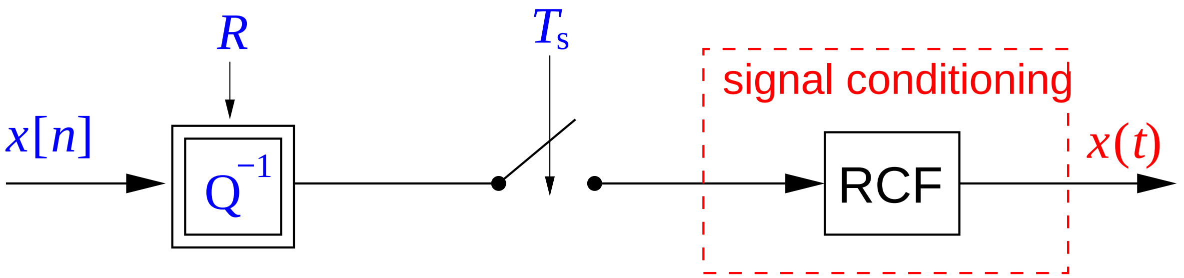
$$\sigma_{xx}^2 = \int_{-V}^{V} x^2 \cdot p(x)dx = \frac{1}{2V}\left[\frac{1}{3}x^3\right]_{-V}^{V} = \frac{V^2}{3} \tag{10}$$

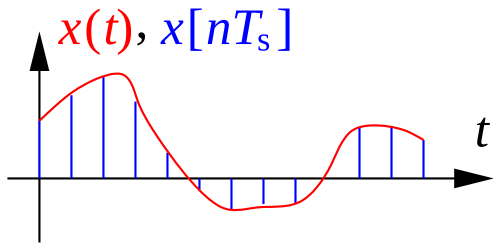
- with (9) and $q = 2V/2^R$, the signal-to-quantisation noise ratio (SQNR) is given by

$$\text{SQNR}_{\text{dB}} = 10 \cdot \log_{10} \frac{V^2 \cdot 12 \cdot 2^{2R-2}}{3 \cdot V^2} = 10 \cdot \log_{10} 2^{2R} \approx 6.02 \cdot R \quad \text{[dB]} \tag{11}$$

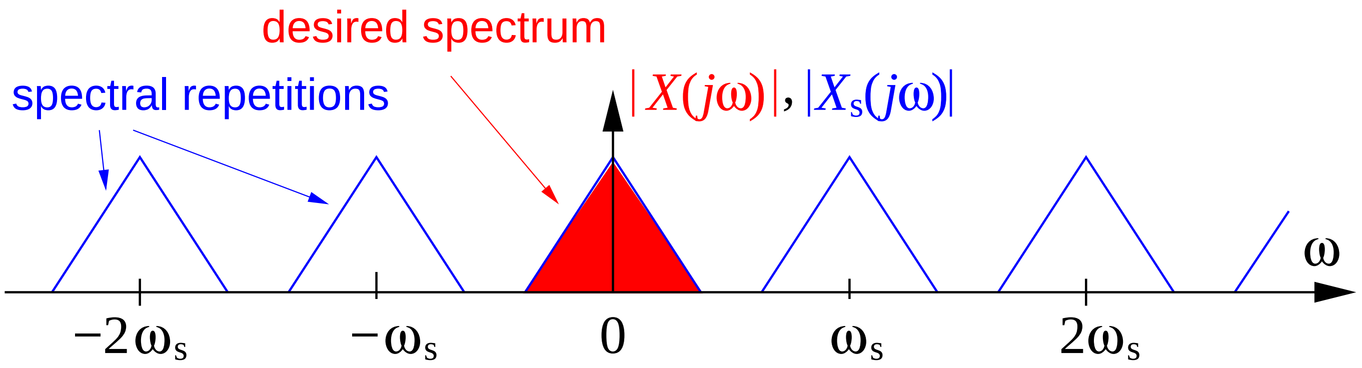
Digital-to-Analogue Conversion

- For the onversion from the digital to the analogue domain, a DAC comprises of:





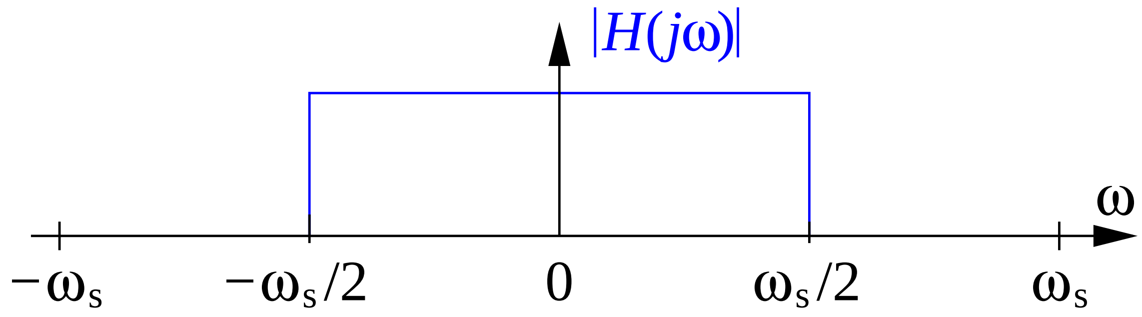
- the voltage signal at the output of Q^{-1} contains high frequency components, which need to be removed:



Reconstruction Filter

- In order to reconstruct the analogue signal $x(t)$, a lowpass filter is required to remove undesired high frequency components;

- this lowpass filter interpolates between adjacent sample values and is therefore termed reconstruction filter; requirement: stopband edge at $\omega_\mathrm{s}/2$:

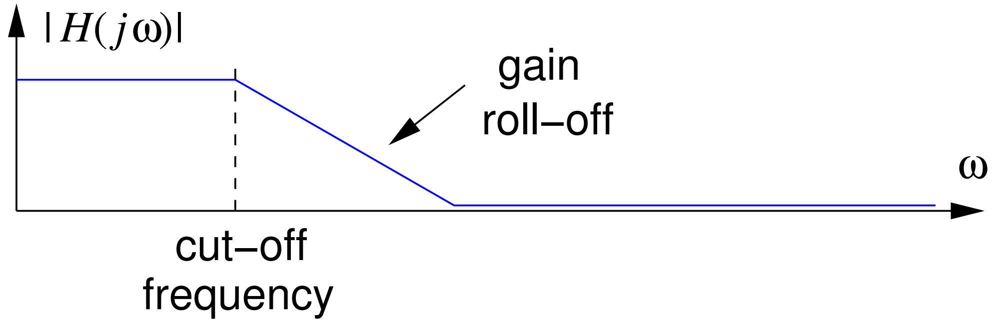


- with a perfect anti-alias filter and reconstruction filter, the concatenation of ADC and DAC imposes no loss on the signal (Nyquist);

- loss occurs due to (i) non-ideal anit-alias and reconstruction filters and (ii) quantisation.

Realistic ADC/DAC

- High quality analogue filters for anti-alias and reconstruction filtering require high order and low tolerances on elements;
- Realistically, a sufficient roll-off has to be permitted, hence limiting the "ideal" reconstruction:



- analogue components can be the most expensive part in a DSP system;
- we will study later how part of the analogue processing can be performed in the digital domain by oversampling and $\Sigma - \Delta$ converters.

Spectrum Estimation

- The discrete Fourier transform (DFT) has been discussed previously (see sections 11 and 12 in Prof. Rogers' notes);

- we want to consider techniques to estimate the spectra of stochastic signals;

- if $\mathbf{x}$ is an N-element vector taken from a tap delay line (TDL) and $\mathbf{T}$ an N-point DFT matrix, then from the transform vector $\mathbf{X}$

$$\mathbf{X} = \mathbf{T} \cdot \mathbf{x} \tag{12}$$

  a *periodogram* can be derived by squaring the Fourier coefficient; problem: if $\mathbf{x}$ is random, then $\mathbf{X}$ will also be random;

- if we want to obtain a deterministic quantity for the spectrum of a random signal, clearly some form of expectation/averaging needs to be incorporated — either in the time or the frequency domain.

Recall: Stochastic Processes

- A stochastic process is characterised by some deterministic measures:
  - its probability density function (PDF) ($=$ normalised histogram);
  - its moments $\int x^l p(x) dx$ of order l;
    (note that $l = 1$ is mean, $l = 2$ is variance for zero-mean process);
  - its autocorrelation function;

- central limit theorem: the sum of arbitrarily distributed processes converges to a Gaussian PDF;

- Gaussian (or normal) PDF of a process $x[n] \in \mathcal{N}(\mu_x, \sigma_{xx}^2)$:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma_{xx}} \, e^{\frac{-(x-\mu_x)^2}{2\sigma_{xx}^2}} \tag{13}$$

Stationarity and Ergodicity

- Stationarity means that the statistical moments of a random process do not change over time;

- a weaker condition is wide-sense stationarity (WSS), i.e. moments up to second order (mean and variance) are constant over time; this is sufficient unless higher order statistics (HOS) algorithms are deployed;

- a stochastic process is ergodic if the expectation operation can be replaced by a temporal average,

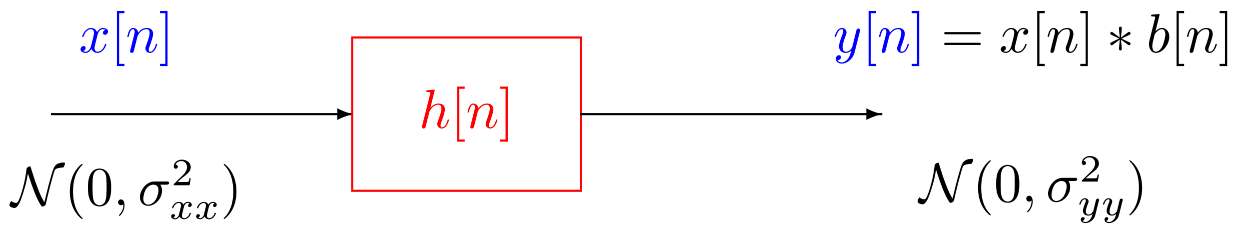
$$\sigma_{xx}^2 = \int_{-\infty}^{\infty} x^2 p(x) dx = \mathcal{E}\{x[n]x^*[n]\} = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} |x[n]|^2 \qquad (14)$$

- remember: expectation is an average over an ensemble; a temporal average is performed over a single ensemble probe!

Moving Average (MA) Model / Signal

- The PDF does not contain any information on how "correlated" successive samples are;

- consider the following scenario with $x[n] \in \mathcal{N}(0, \sigma_{xx}^2)$ being uncorrelated (successive samples are entirely random):



- $y[n]$ is called a moving average process (and $b[n]$ an MA model) of order $N-1$ if $y[n] = \sum_{\nu=0}^{N-1} b[\nu]x[n-\nu]$ is a weighted average over a window of N input samples.

Autoregressive (AR) and ARMA Models / Signals

- If the process $y[n]$ has emerged from an uncorrelated process $x[n] \in \mathcal{N}(0, \sigma_{xx}^2)$ via the difference equation

$$y[n] = b[0]x[n] - \sum_{\nu=1}^{N-1} a[\nu]y[n-\nu] \quad ; \tag{15}$$

it is called an $N-1$ order auroregressive ("self-dependent",[2]) signal;

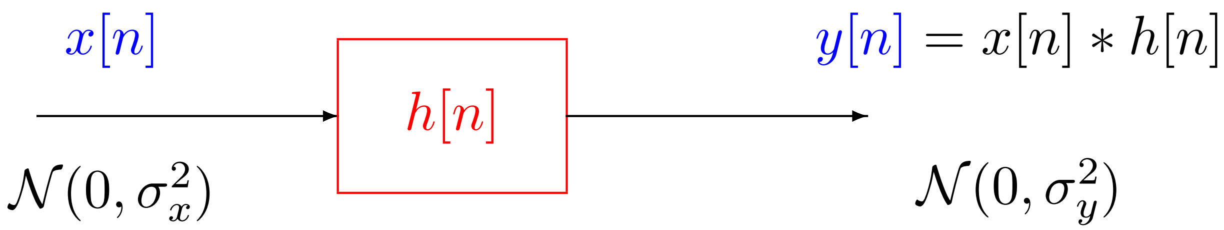
- the system associated with (15), defined by its parameters $b[0]$ and $a[\nu]$, $\nu = 1 \cdots N-1$, is referred to as an autoregressive model.

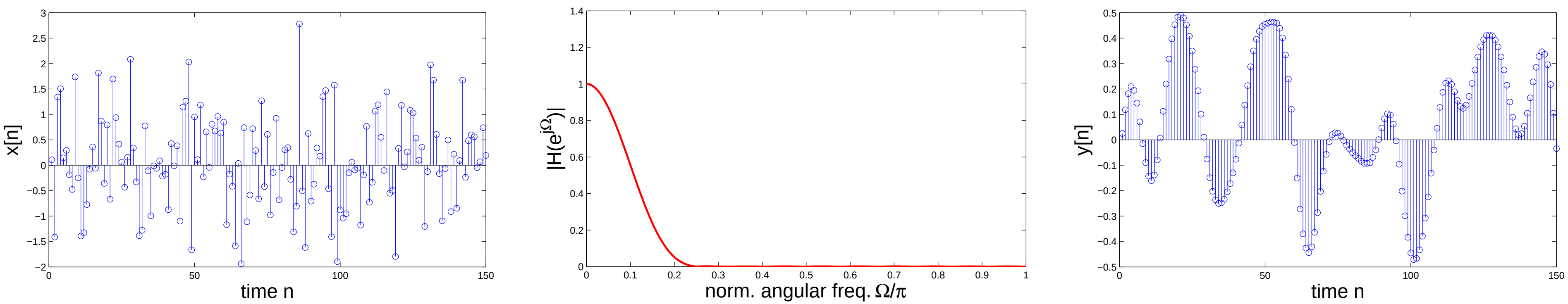
- an MA model is often also called an all-zero model or FIR filter; an AR model is an all-pole filter; autoregressive moving average (ARMA) systems refer to a general IIR filter;

- such a model can be very useful in characterising the stochastic process $y[n]$ by an *innovations filter*[3].

Filtering a Random Signal

- Consider lowpass filtering an uncorrelated Gaussian signal $x[n]$:





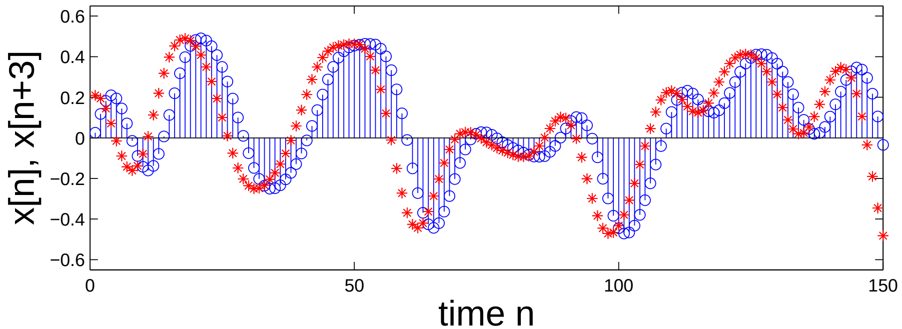
- the output will have Gaussian distribution, but the signal only changes smoothly: neighbouring samples are "correlated". We need a measure!

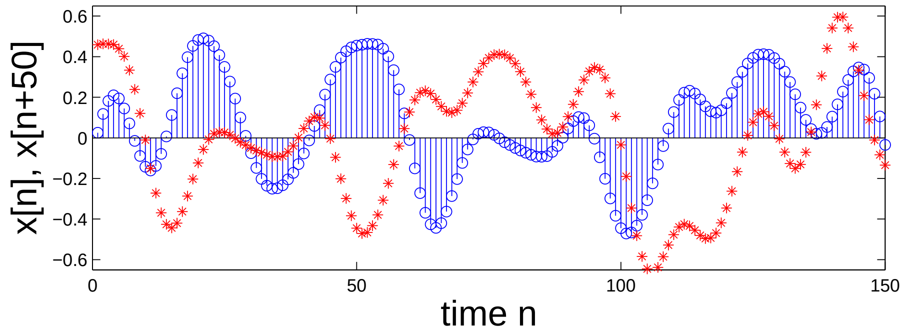
Auto-Correlation Function I

- The correlation between a sample $x[n]$ and a neighbouring value $x[n-\tau]$ is given by

$$r_{xx}[\tau] = \mathcal{E}\{x[n] \cdot x^*[n-\tau]\} = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} x[n] \cdot x^*[n-\tau] \qquad (16)$$

- For two specific specific lags $\tau = 3$ (left) and $\tau = 50$ (right), consider:





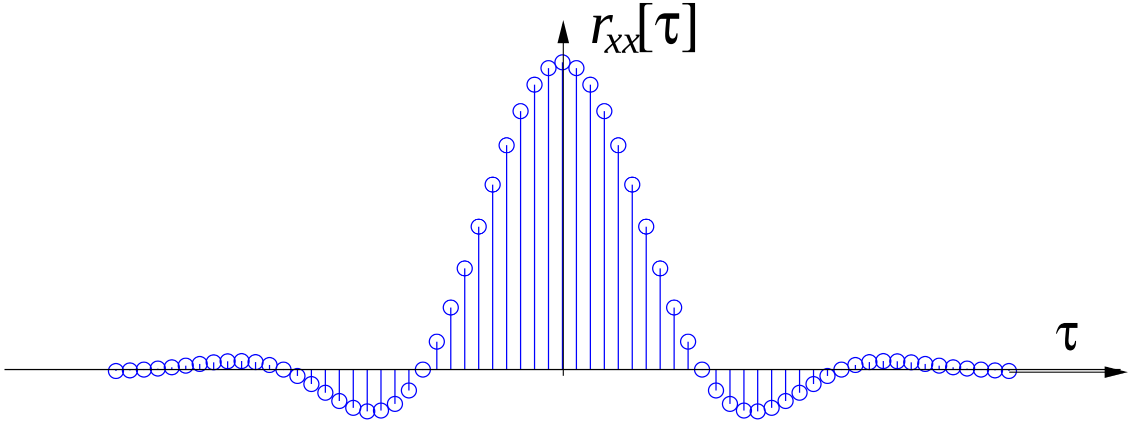
- the curves on the left look “similar”, the ones on the right “dissimilar”.

Auto-Correlation Function II

- For lag zero, note:

$$r_{xx}[0] \;=\; \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} x[n] \cdot x^*[n] = \sigma_x^2 + \mu_x^2 \qquad (17)$$

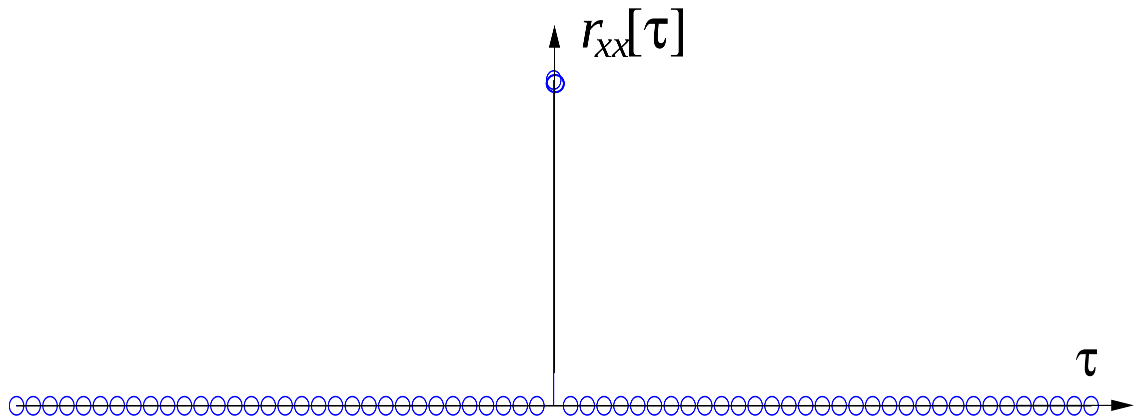
- This value for $\tau = 0$ is the maximum of the auto-correlation function $r_{xx}[\tau]$;



- large values in the ACF indicate strong correlation, small values weak correlation;

Auto-Correlation Function III

- If a signal has no self-similarity, i.e. it is "completely random", the ACF takes the following form:



- If we take the Fourier transform of $r_{xx}[\tau]$, we obtain a flat spectrum (or a lowpass spectrum for the ACF on slide 28);

- due to the presence of all frequency components in a flat spectrum, a completely random signal is often referred to as "white noise".

Power Spectral Density

- The power spectral density (PSD), $R_{xx}(e^{j\Omega})$, defines the spectrum of a random signal:

$$R_{xx}(e^{j\Omega}) = \sum_{\tau=-\infty}^{\infty} r_{xx}[\tau]\, e^{j\Omega\tau} \tag{18}$$

- PSD and ACF form a Fourier pair, $r_{xx}[\tau] \circ\!\!-\!\!\bullet\, R_{xx}(e^{j\Omega})$, therefore

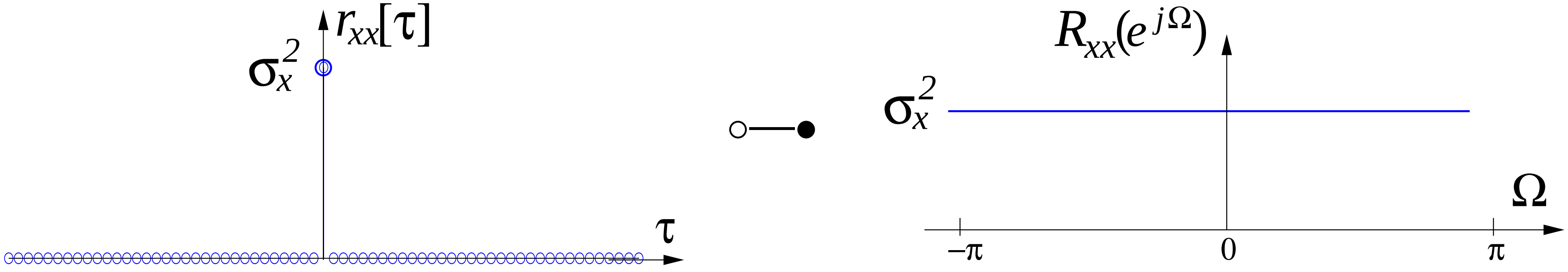
$$r_{xx}[\tau] = \frac{1}{2\pi}\int_{-\pi}^{\pi} R_{xx}(e^{j\Omega})\, e^{-j\Omega\tau}\, d\Omega \tag{19}$$

- note that the power of $x[n]$ is (similar to Parseval)

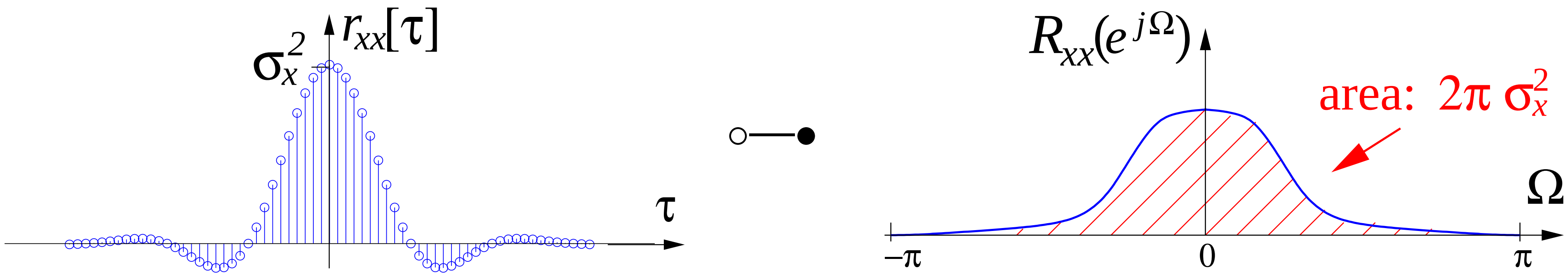
$$r_{xx}[0] = \frac{1}{2\pi}\int_{-\pi}^{\pi} R_{xx}(e^{j\Omega})\, d\Omega \qquad (= \text{scaled area under PSD}) \tag{20}$$

PSD – Examples

- PSD for uncorrelated ("white") zero mean noise:



- PSD for correlated zero mean noise:



Cross-Correlation

- The cross-correlation function between two signal $x[n]$ and $y[n]$ is defined analogous to (16):

$$r_{xy}[\tau] \;=\; \mathcal{E}\{x[n] \cdot y^*[n-\tau]\} \;=\; \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} x[n] \cdot y^*[n-\tau] \qquad (21)$$

- note: $r_{yx}[\tau] \;=\; r^*_{xy}[-\tau]$ and especially $r_{xx}[\tau] \;=\; r_x x[-\tau]$

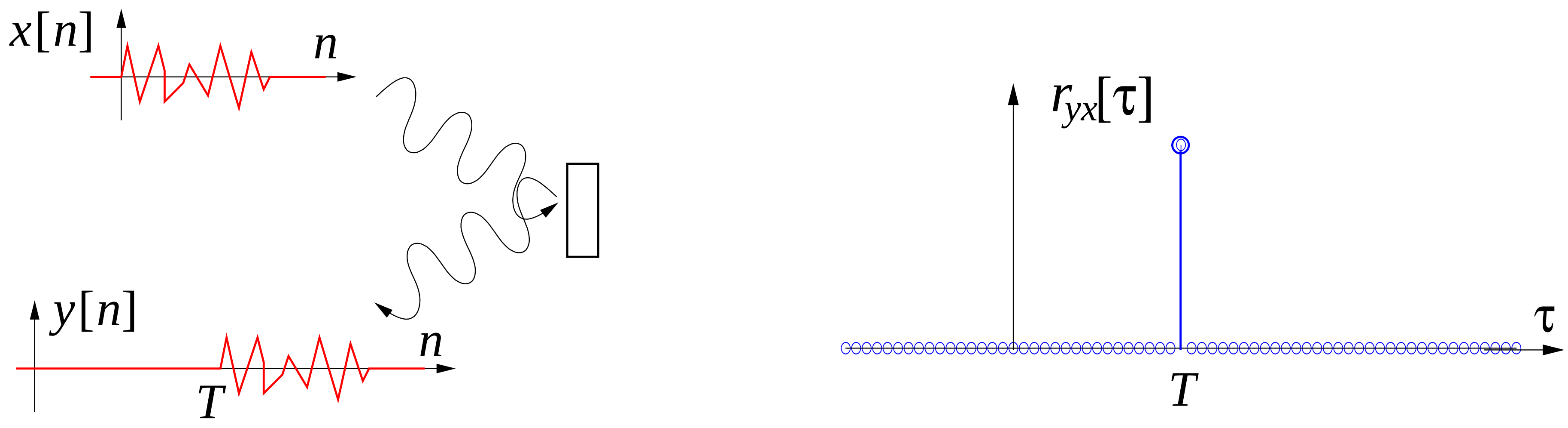
  i.e. the auto-correlation function is symmetric, while the cross-correlation function is not;

- for uncorrelated signals:

$$r_{xy}[\tau] \;=\; \mathcal{E}\{x[n] \cdot y^*[n-\tau]\} \;=\; \mathcal{E}\{x[n]\} \cdot \mathcal{E}\{y^*[n-\tau]\} \;=\; \mu_x \, \mu_y^* \qquad (22)$$

Examples for Applying the Cross-Correlation

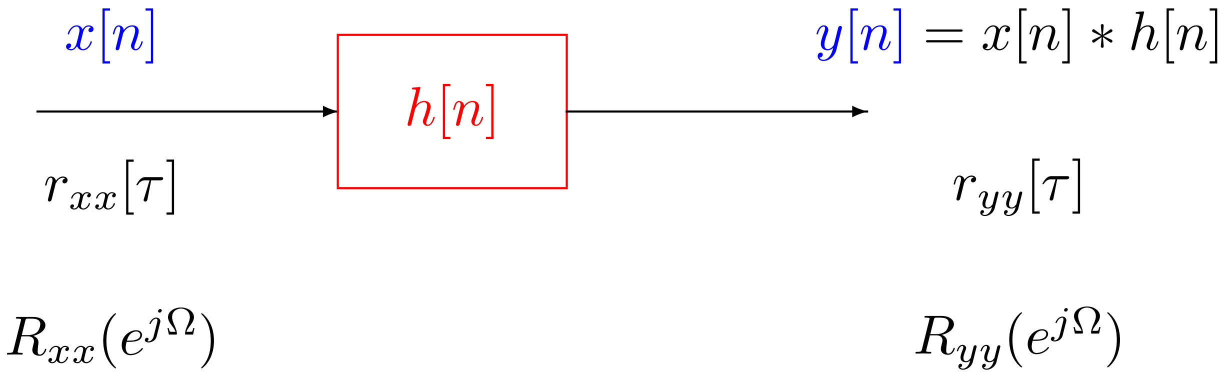
- Delay estimation. Imagine we send a random pulse $x[n]$ and wait for the reflected return signal $y[n]$:



- "Matched filtering". Compare a received signal with an expected waveform; the cross-correlation will be maximum if the received signal matches up with a desired sequence.

PSD and Innovations Filter

- Consider again filtering a random signal $x[n]$ with a filter having an impulse response $h[n]$ (either MA, AR, or ARMA):



- relation between $x[n]$ and $y[n]$ is given by convolution: $y[n] = \sum_{\nu=-\infty}^{\infty} h[\nu]\, x[n-\nu]$;

- we are looking for the relations between $r_{xx}[\tau]$ and $r_{yy}[\tau]$ and between $R_{xx}(e^{j\Omega})$ and $R_{yy}(e^{j\Omega})$.

- The cross-correlation is:

$$r_{yx}[\tau] = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} y[n] \cdot x^*[n-\tau] \quad (23)$$

$$= \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} \cdot \left(\sum_{\nu=-\infty}^{\infty} h[\nu]\, x[n-\nu] \right) x^*[n-\tau] \quad (24)$$

$$= \sum_{\nu=-\infty}^{\infty} h[\nu] \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} x[n-\nu]\, x^*[n-\tau] \quad (25)$$

$$= \sum_{\nu=-\infty}^{\infty} h[\nu] \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} x[n]\, x^*[n-\tau+\nu] \quad (26)$$

$$= \sum_{\nu=-\infty}^{\infty} h[\nu]\, r_{xx}[\tau-\nu] = h[\tau] * r_{xx}[\tau] \quad (27)$$

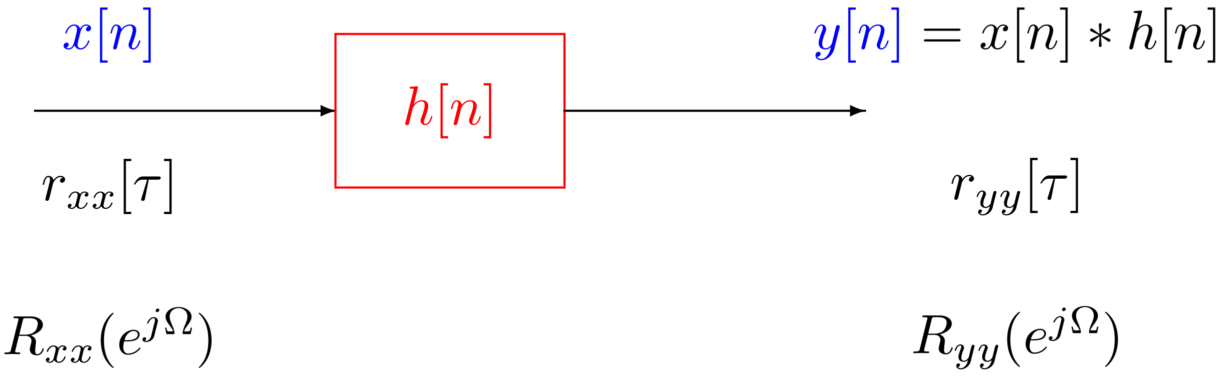
- note: $r_{xy}[\tau] = r^*_{yx}[-\tau] = h^*[-\tau] * r_{xx}[\tau]$

- Going further:

$$
\begin{aligned}
r_{yy}[\tau] &= \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} y[n] \cdot y^*[n-\tau] && (28)\\
&= \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} y[n] \sum_{\nu=-\infty}^{\infty} h^*[\nu]\; x^*[n-\nu] && (29)\\
&= \sum_{\nu=-\infty}^{\infty} h^*[\nu] \;\lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} y[n]\; x^*[\nu-n+\tau] && (30)\\
&= \sum_{\nu=-\infty}^{\infty} h^*[\nu]\; r_{yx}[\nu+\tau] \;=\; h^*[-\tau] * r_{yx}[\tau] && (31)\\
&= h^*[-\tau] * h[\tau] * r_{xx}[\tau] && (32)
\end{aligned}
$$

PSD and Innovations Filter II

- Hence, if a stochastic process $x[n]$ is filtered:



- we have

$$r_{yy}[\tau] = h^*[-\tau] * h[\tau] * r_{xx}[\tau] \quad (33)$$

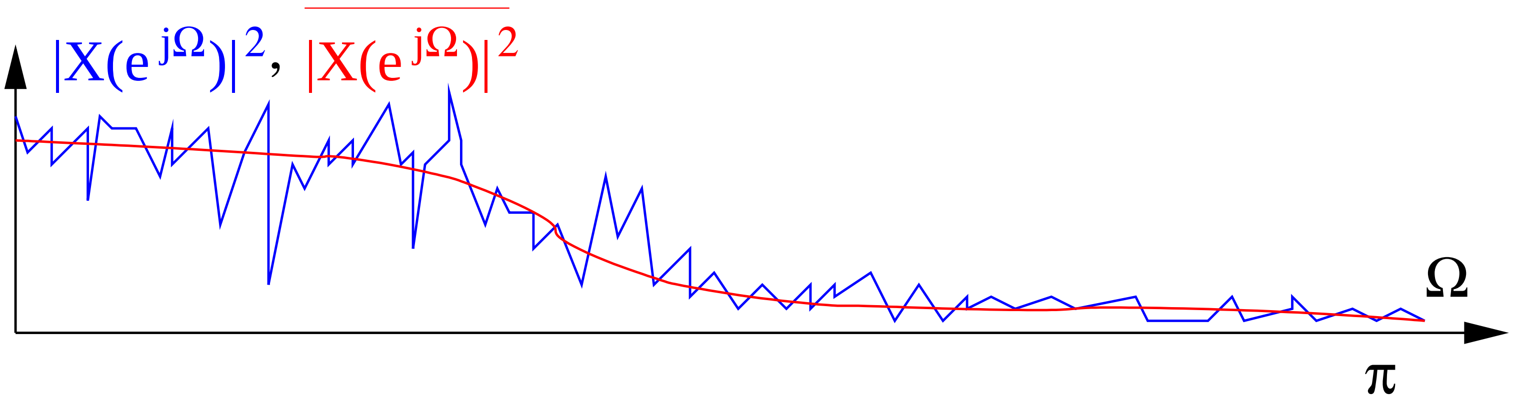
$$R_{yy}[\tau] = H^*(e^{j\Omega})\; H(e^{j\Omega})\; R_{xx}[\tau] \quad (34)$$

$$= |H(e^{j\Omega})|^2\; R_{xx}[\tau] \quad (35)$$

- note for innovations filter (uncorrelated input $x[n] \in \mathcal{N}(0,1)$): $R_{yy}[\tau] = |H(e^{j\Omega})|^2$.

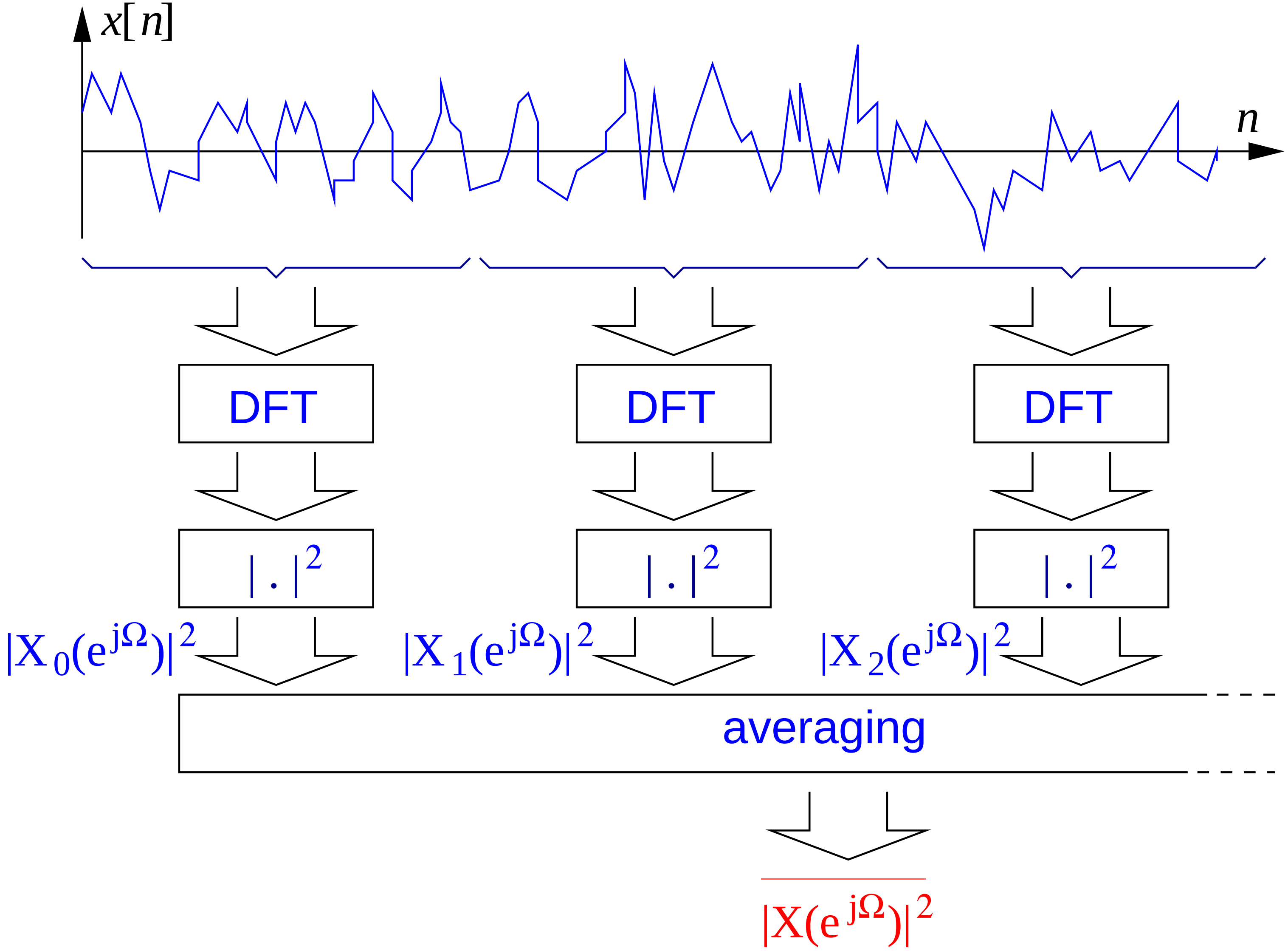
Averaged Periodogram I

- The PSD is a deterministic quantity of a random process, while the periodogram is random;
- averaging of the periodogram is required to eliminate (or at least reduce randomness);
- method 1: take N-point DFT of $x[n]$, and average over a neighbourhood of squared Fourier coefficients;



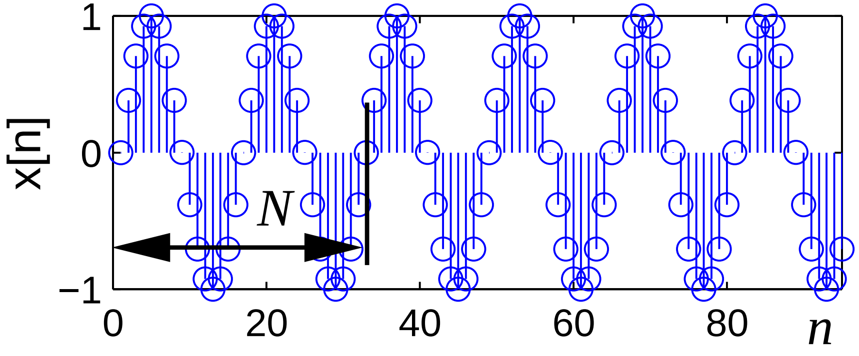
- method 2: calculate a number of periodograms and average across the the frequency bins (Fourier coefficients).

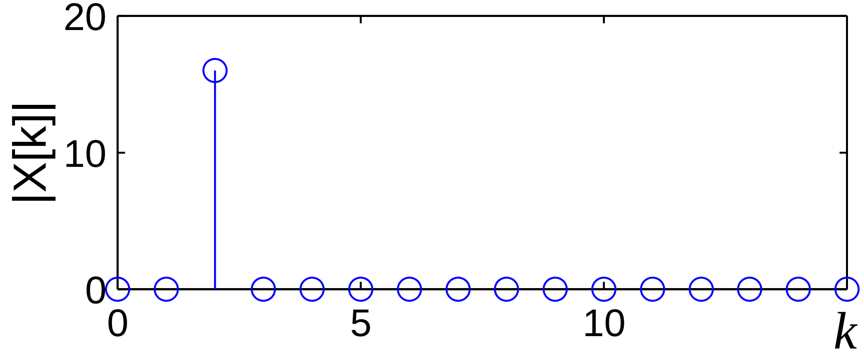
Averaged Periodogramme II



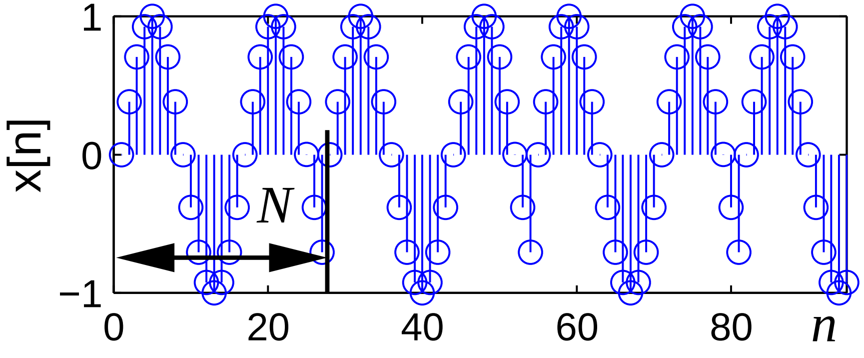
Recall Spectral Leakage

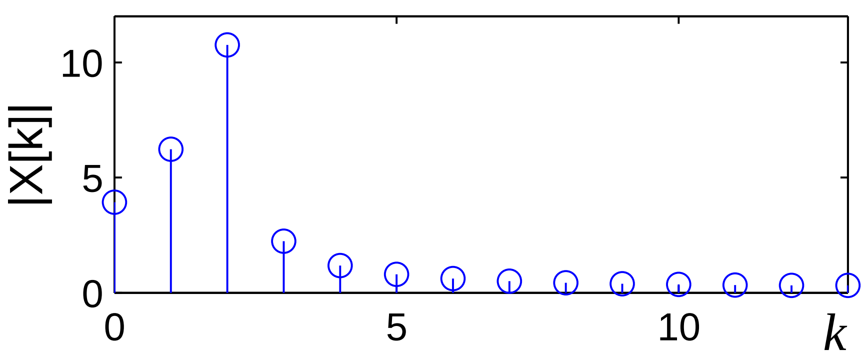
- The DFT "periodises" the data, which is likely to create aberrations;
- $N = 32$, by chance the window ends fit:





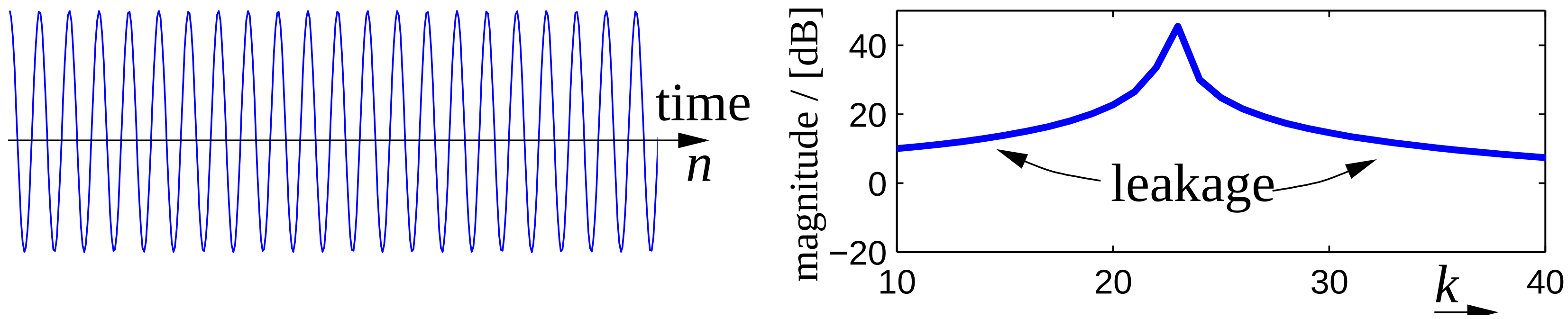
- $N = 27$, discontinuties arise at the window edges, causing the main peak to "leak":



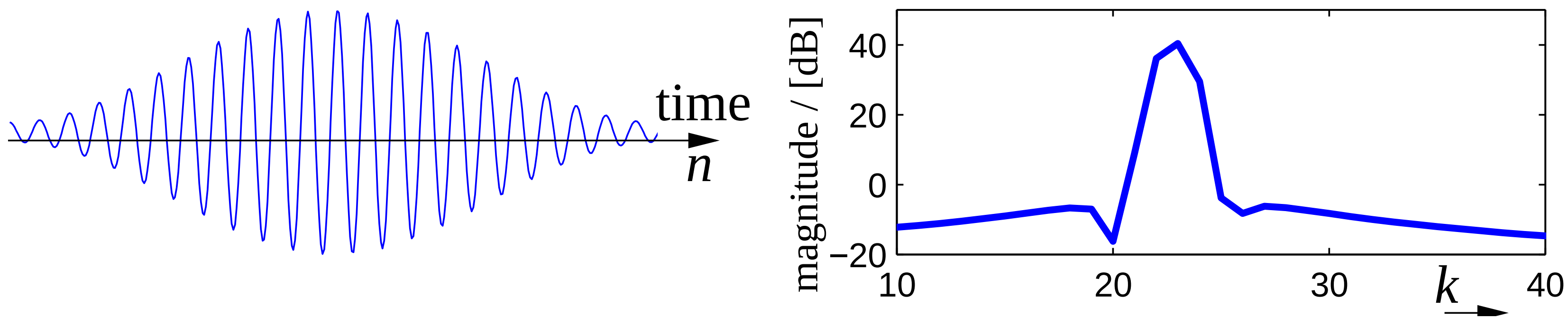


Recall Windowing

- DFT applied to a segment of rectangularly windowed sinusoidal data:

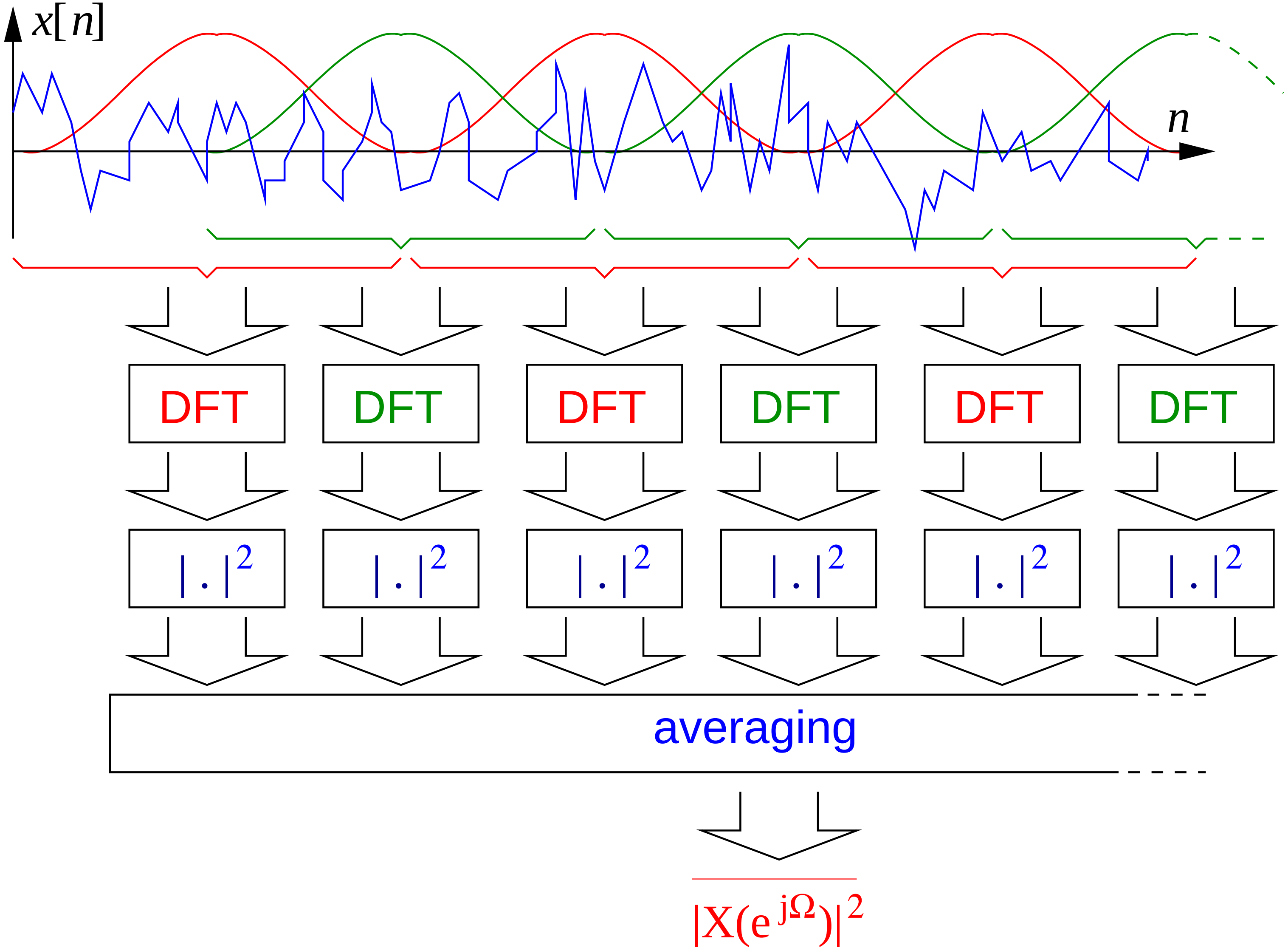


- DFT after application of a Hamming window to the same data:



- with windowing, the spectral leakage is reduced at the cost of a widened main lobe.

Practical PSD Calculation



Relation between Averaged Periodogram and PSD

- Note the differences and correspondences between the PSP and the averaged periodogram:

$$\begin{array}{ccccccc} & & & \mathcal{E}\{\cdot\} & & & \\ & & x[n] & \longrightarrow & r_{xx}[\tau] & & \\ \text{DFT} & & \downarrow & & \downarrow & & \text{DFT} \\ & & |X(e^{j\Omega})|^2 & \longrightarrow & R_{xx}(e^{j\Omega}) & & \\ & & & \mathcal{E}\{\cdot\} & & & \end{array} \tag{36}$$

- spectral estimation (approximations of the PSD) are usually based on the averaged periodogram method outlined in slide 42.

Summary

- We have defined PSD theoretically and the averaged periodogram as a computational technique to estimate it;
- usually the longer the data sequence $x[n]$ available, the more confident (accurate) this estimate is;
- we will define a parametric method for spectrum estimation, trying to guess at the innovations filter behind the signal at hand;
- for background and further reading:

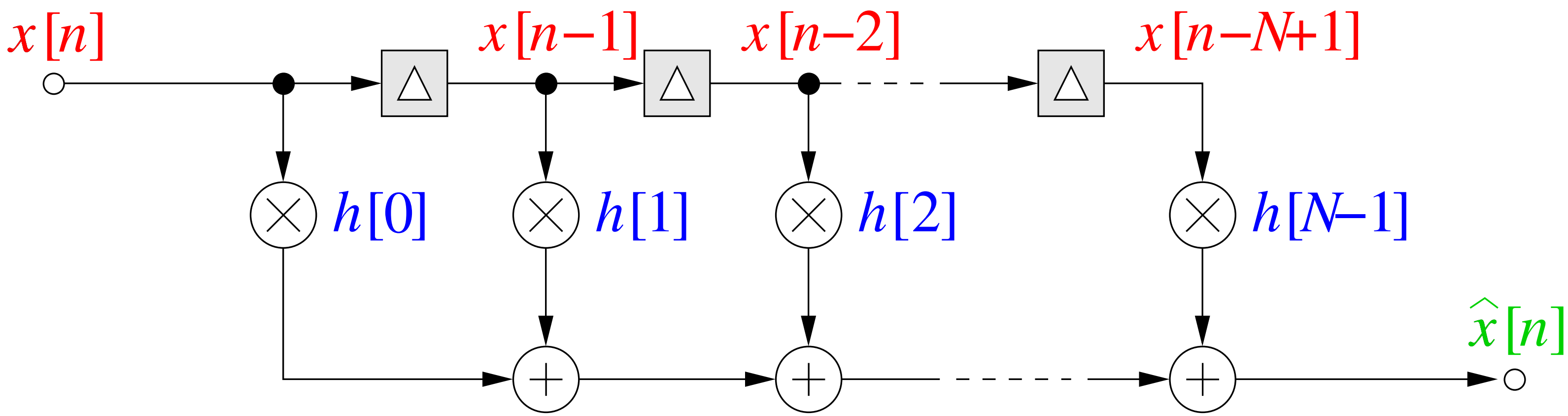
[1] B. Girod, R. Rabenstein, and A. Stenger. *Signals and Systems*. J. Wiley & Sons, 2001. *[Good explanation of auto- and cross-correlation, and the PSD]*

[2] T. K. Moon and W. C. Stirling. *Mathematical Methods and Algorithms for Signal Processing*. Prentice Hall, 1999.

[3] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 3rd ed., 1991. *[A classic text.]*

Digital FIR Filtering

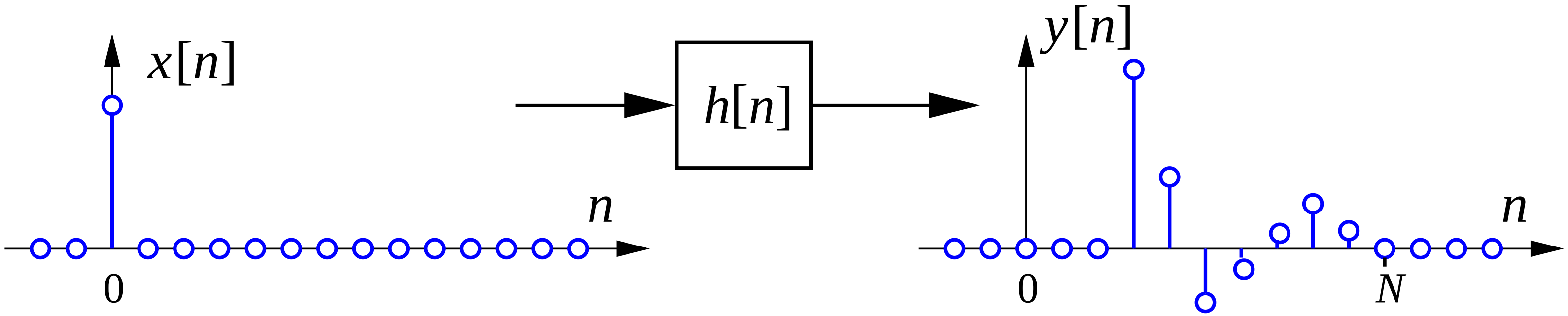
- Finite impulse response (or moving average, MA) filter:



- tap delay line (TDL) of length N;
- described by difference equation, $y[n] = \sum_{\nu=0}^{N-1} b_n x[n-\nu]$
- no feedback: inherently stable.

Impulse Response

- For an FIR filter, the impulse response $h[n]$ (earlier stated as zero-state response, i.e. the TDL is initialised to zero) is given by its filter coefficients:

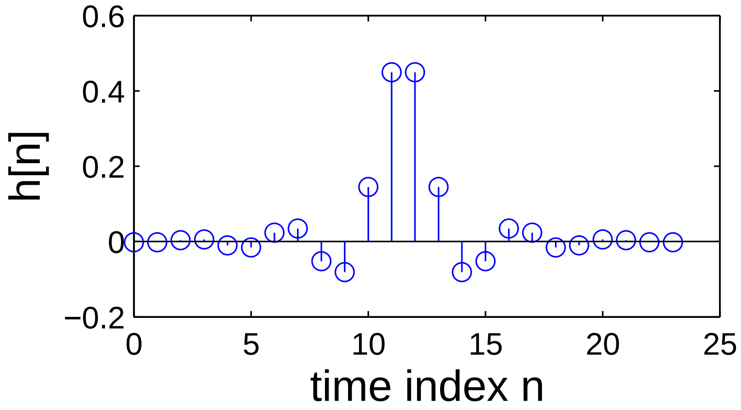


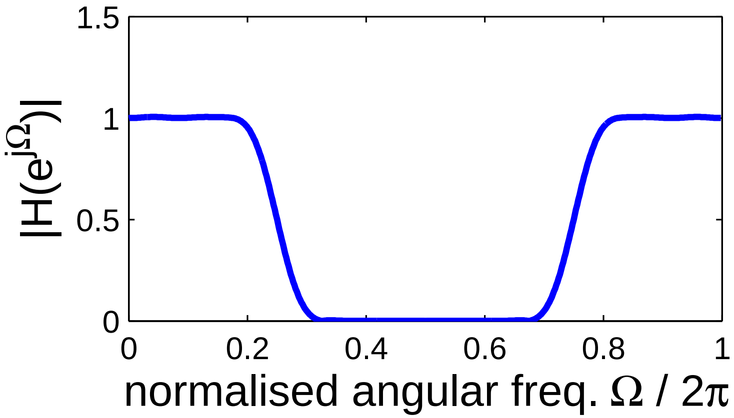
- the impulse response has finite support with $h[n] = 0 \;\; \forall n \geq N$;

- the impulse response can give direct clues about multipath propagation, or echos and reverberation.

Frequency Response

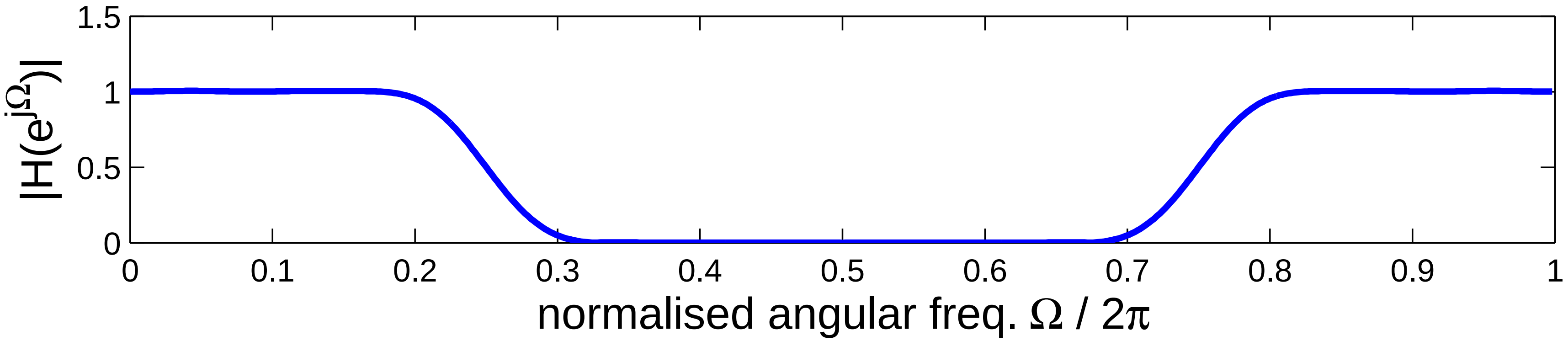
- Frequency response $H(e^{j\Omega})$: measures the gain of the system in the *steady state*:
- impulse and frequency response are a Fourier pair: $h[n] \circ\!\!-\!\!\bullet H(e^{j\Omega})$.



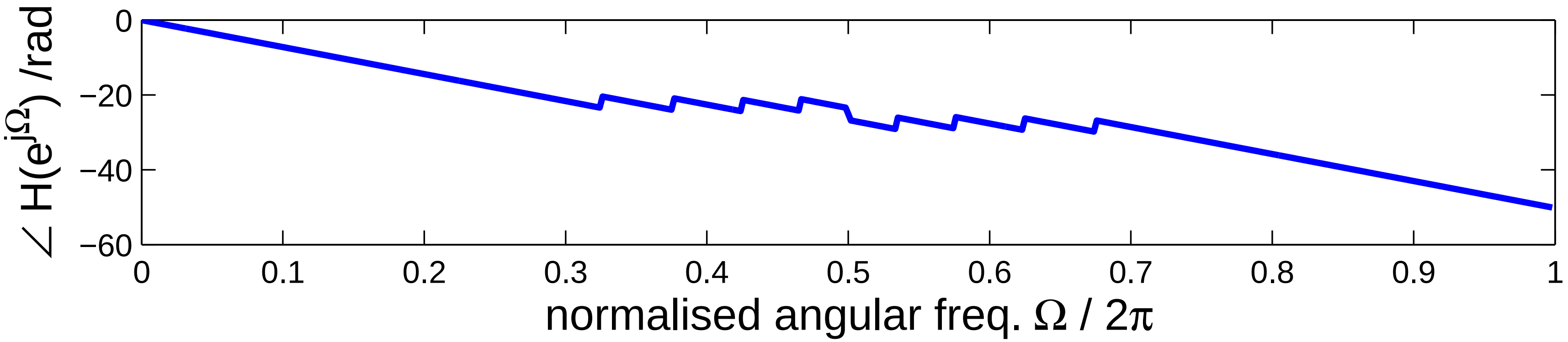


Magnitude and Phase Response

- The magnitude decides whether the system has e.g. lowpass, highpass, or bandpass characteristic;



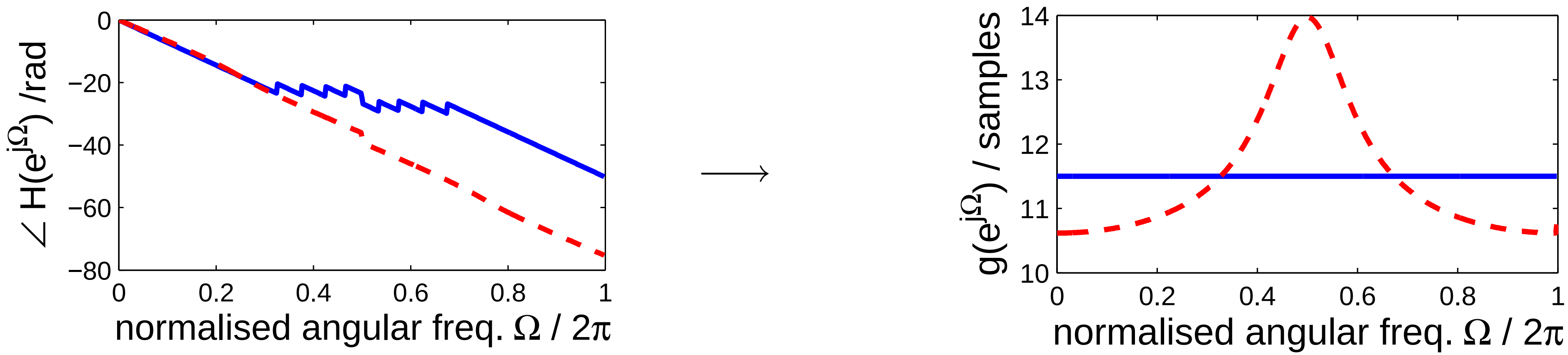
- phase contains a number of exciting characteristics of a filter:



Group Delay

- from $H(e^{j\Omega}) = |H(e^{j\Omega})|e^{j\Phi(\Omega)}$, note for the group delay $g(\Omega)$:

$$g(e^{j\Omega}) = -\frac{d}{d\Omega}\Phi(\Omega) \tag{37}$$



- the group delay tells us by how long certain frequency components will be delayed when propagating through the filter;

- note the difference between linear phase and non-linear phase systems.

z-Transform

- Reconsider FIR filter:

- z-transform of difference equation:

$$Y(z) = b_0X(z) + b_1z^{-1}X(z) + b_2z^{-2}X(z) + \cdots + b_{N-1}z^{-N+1}X(z) \quad (38)$$

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{i=0}^{N-1} b_i z^{-i} \quad (39)$$

- the z-transform with $z = e^{j\Omega+\alpha}$ can capture the transient behaviour of a system;

- note that $H(e^{j\Omega}) = H(z)|_{z=e^{j\Omega}}$.

z-Transform — Some Useful Properties

- Assume a z-transform pair $h[n] \bullet\!\!-\!\!\circ\; H(z) = \sum_{i=-\infty}^{\infty} h_i \, z^{-i}$

- time shift:

$$h[n-\Delta] \quad \bullet\!\!-\!\!\circ \quad H(z)\, z^{-\Delta} \tag{40}$$

- complex conjugation:

$$h^*[n] \quad \bullet\!\!-\!\!\circ \quad H^*(e^{j\Omega}) = \sum_{i=-\infty}^{\infty} h_i^* \, z^{-i} \tag{41}$$

- time reverse:

$$h[-n] \quad \bullet\!\!-\!\!\circ \quad H(z^{-1}) = \sum_{i=-\infty}^{\infty} h_i \, z^{i} \tag{42}$$

Transfer Function

- We can factorise the transfer function $H(z)$:

$$H(z) = b_0 + b_1 z^{-1} + \cdots b_{N-1} z^{-N+1} \quad (43)$$

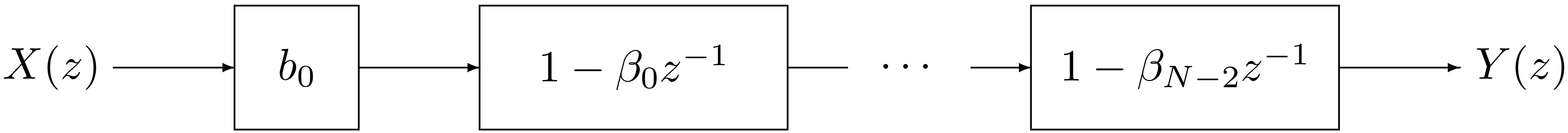
$$= b_0(1 - \beta_0 z^{-1})(1 - \beta_1 z^{-1}) \cdots (1 - \beta_{N-2} z^{-N+2}) \quad (44)$$

- the roots β_i of $H(z)$ are called the zeros of the transfer function;
- often a pole-zero plot is used for visulisation:

Real Valued Systems

- If the impulse response only contains real-valued coefficients, the zeros β_i must be either real valued, or occur as complex conjugate pairs (as seen in slide52;

- we could factorise the system into a sequence of first order sections:

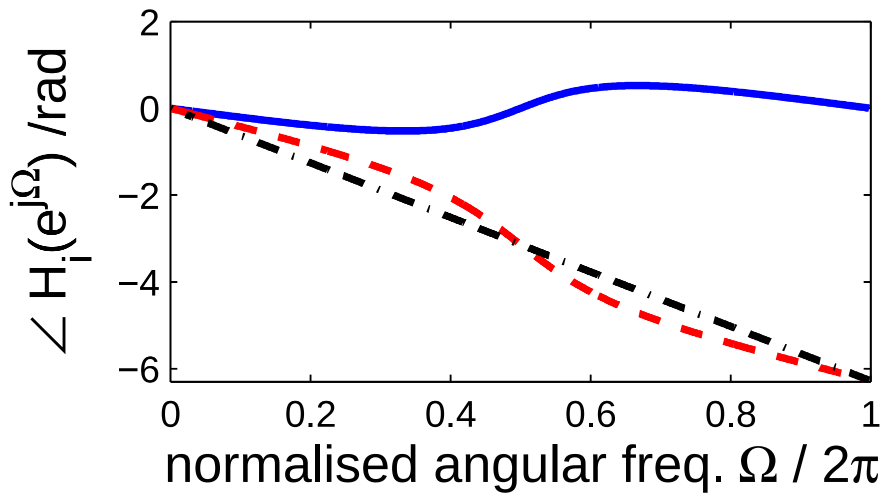


- a complex multiplication take 4 real valued multiplications; therefore, for implementational purposes, we would always prefer (44) over (44).

Minimum and Maximum Phase

- For a minimum phase system, all zeros lie inside the unit circle;
- a maximum phase system has all zeros outside the unit circle;
- interesting: if $h[n]$ is minimum phase, then $h[-n]$ is maximum phase;
- example for $H_1(z) = 1 + \frac{1}{2}z^{-1}$ and $H_2(z) = \frac{1}{2} + z^{-1}$:

$H_1(z)$
$H_2(z)$
$H_3(z) = H_1(z)H_2(z)$



Linear Phase Filters

- recall: linear phase means constant group delay;
- a linear phase system must have a z-transform of the form (including a possible delay for causality)

$$H(z) = H_{\text{min. phase}}(z)\, H_{\text{min. phase}}(z^{-1})\, \Pi_m (1 - e^{j\phi_m} z^{-1}) \tag{45}$$

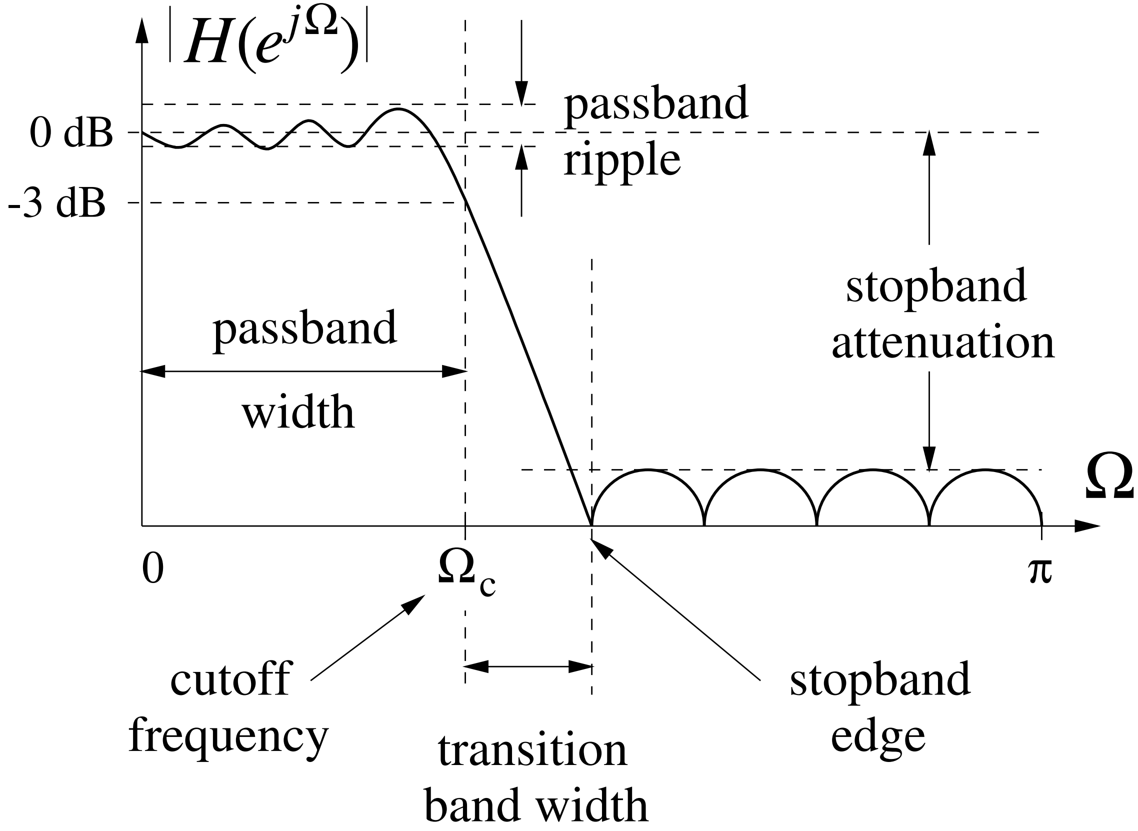
- any zero of a linear phase system must have a reciprocially matching partner (unless on the unit circle);
- linear phase iff impulse response is symmetric: recall that $h_{\text{min. phase}}[n] * h_{\text{min. phase}}[-n]$ will give an auto-correlation type result.

Summary

- Note that certain properties of a system are best assessed in a specific domain: either impulse response, frequency domain, or transfer function;
- phase information is often vital, e.g. if we need to invert the system (for equalisation or control purposes);
- most FIR filter designs yield symmetric impulse response and hence are linear phase;
- we will consider such filter designs next ...

FIR and IIR Design

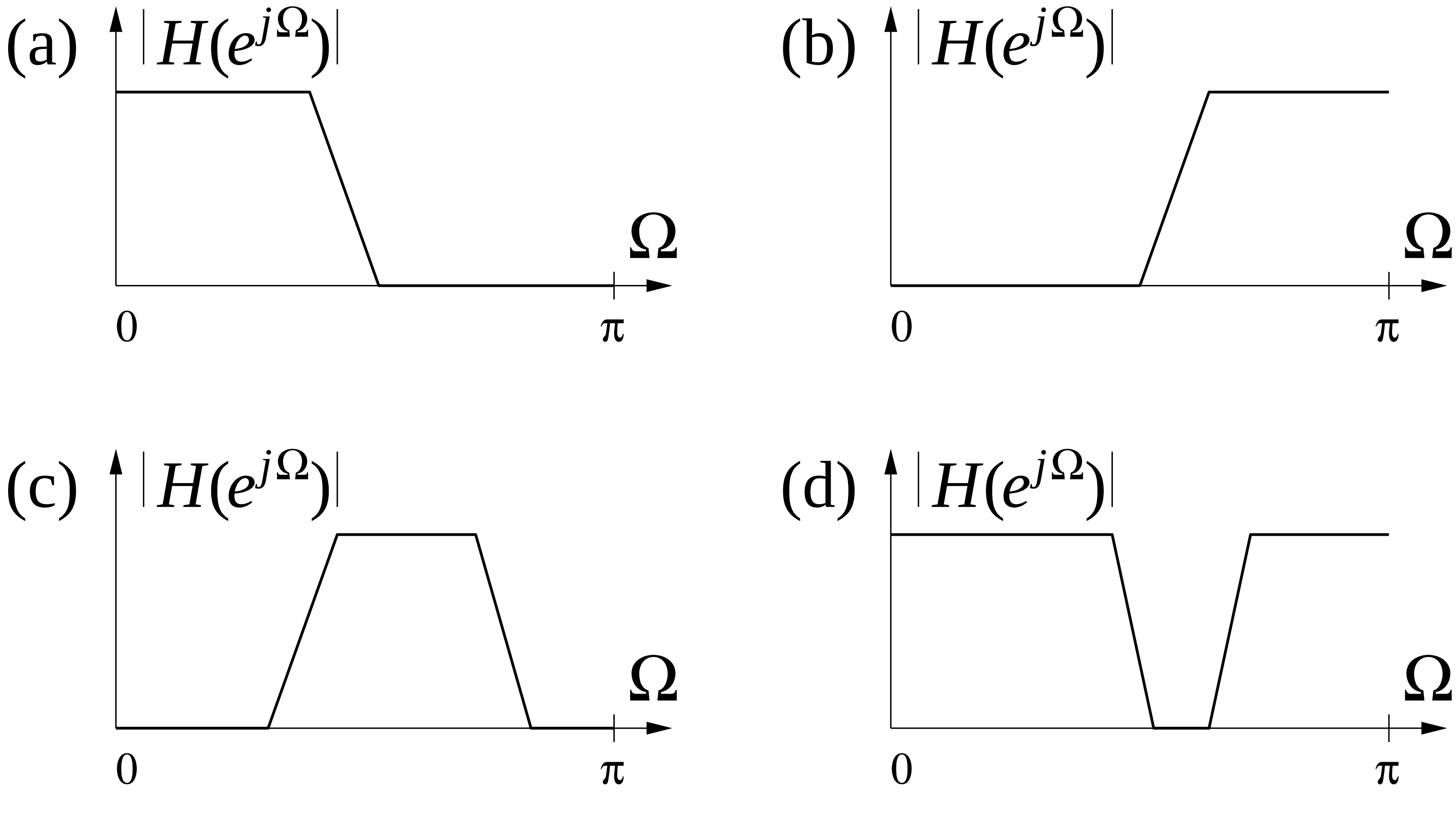
- Design characteristics: passband width, transition band width, stopband edge, stopband attenuation, and passband ripple:



- generally: high quality requires high number of coefficients.

Different Frequency Responses

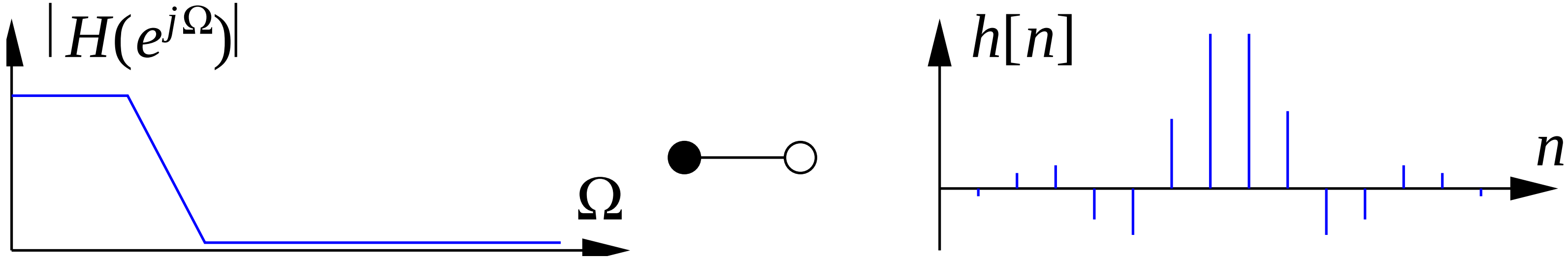
- Four different basic filter types as defined by their magnitude response: (a) lowpass, (b) highpass, (c) bandpass, and (d) bandstop filter:



FIR Filter Design: IDFT and Windowing

- We often have an idea what the magnitude response $|H(e^{j\Omega})|$ of the desired filter should look like; additionally, we need to "invent" a phase response (e.g. linear);

- the frequency response is inversely Fourier transformed;



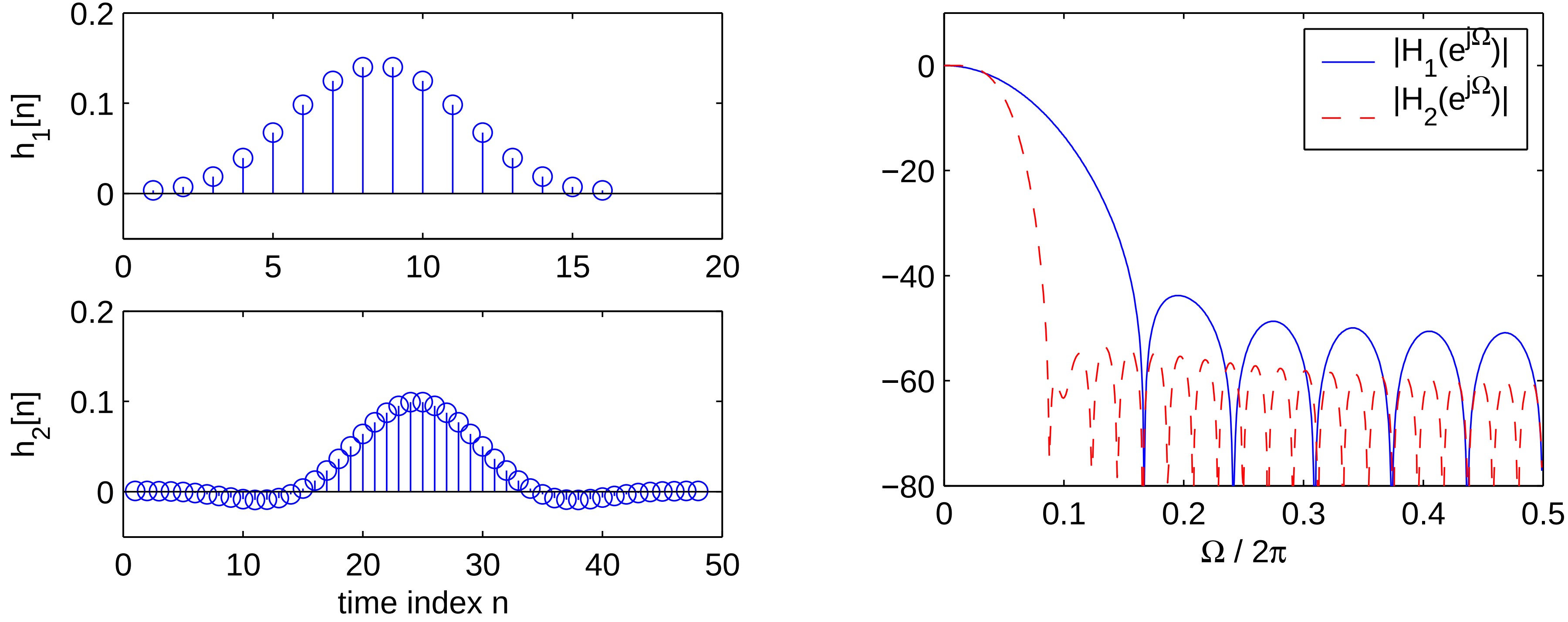
- the resulting time domain response is an approximation of the desired impulse response (holding the filter coefficients);

- some windowing may be required in order to enforce finite support in the time domain.

- a variety of design approaches based on minimax or least squares methods exists.

FIR Design Examples

- Comparison of two filter designs with 16 and 48 coefficients, respectively:

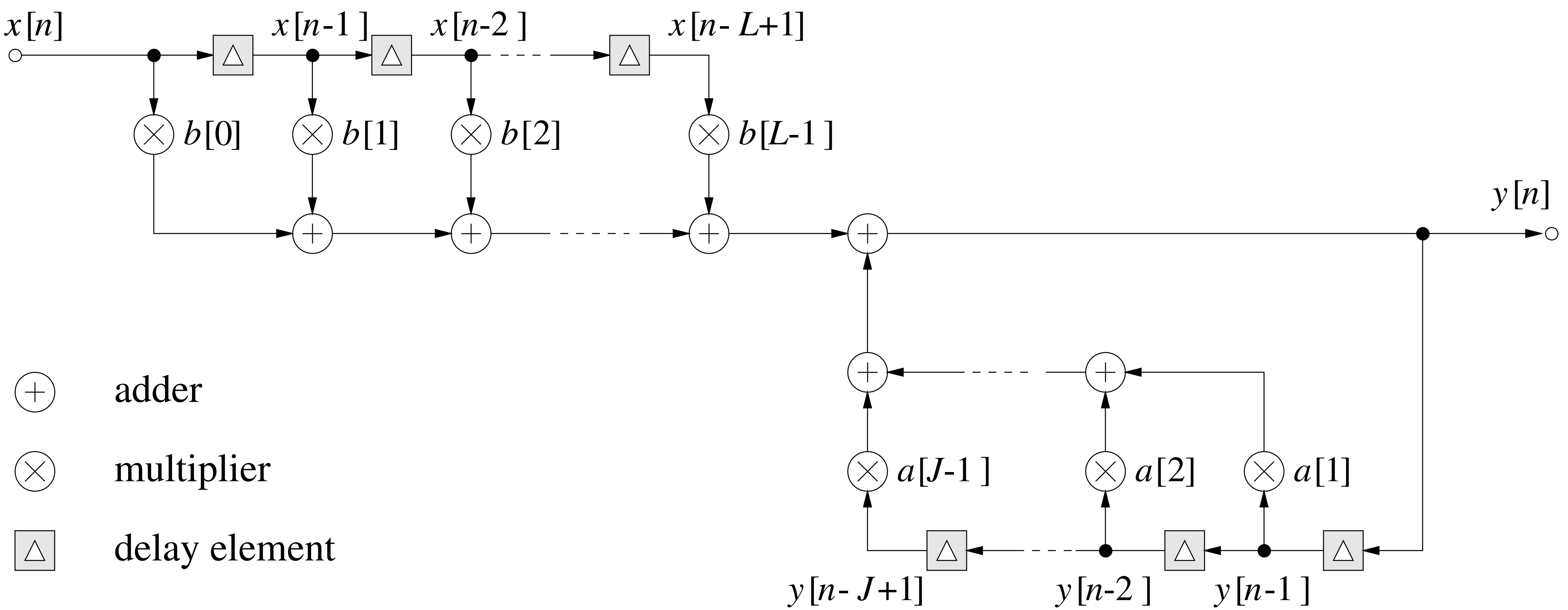


- both designs have identical passband edge at $\Omega = 0.1\pi$.

IIR Filtering

- Various ways to represent the difference equation as a flow graph:

$$y[n] = \sum_{l=0}^{L-1} a_l \, x[n-l] \; + \; \sum_{j=1}^{J-1} b_j \, y[n-j] \tag{46}$$



IIR Stability

- transfer function:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + \cdots b_{L-1} z^{-L+1}}{1 - a_1 z^{-1} - a_2 z^{-2} - \cdots - a_{J-1} z^{-J+1}} \tag{47}$$

$$= b_0 \frac{(1-\beta_0 z^{-1})(1-\beta_1 z^{-1}) \cdots (1-\beta_{L-2} z^{-1})}{(1-\alpha_0 z^{-1})(1-\alpha_1 z^{-1}) \cdots (1-\alpha_{J-2} z^{-1})} \tag{48}$$

- : note that $H(z)$ can be spit into a cascade of FIR and recursive first-order sections;

- each recursive section $\frac{1}{1-\alpha z^{-1}}$ must be stable; note geometric series

$$\frac{1}{1-\alpha z^{-1}} = \sum_{i=0}^{\infty} \alpha^i \, z^{-i} \qquad \text{for} \quad |a| < 1 \tag{49}$$

- poles must be inside the unit circle!

IIR Filter Design

- IIR filter design is based on analogue techniques (Butterworth, Chebychev etc);

- the analogue design is transformed from the Laplace- to the z-domain by means of the bilinear transform:

$$s = \frac{1}{T_\mathrm{s}} \frac{1 - z^{-1}}{1 + z^{-1}} \tag{50}$$

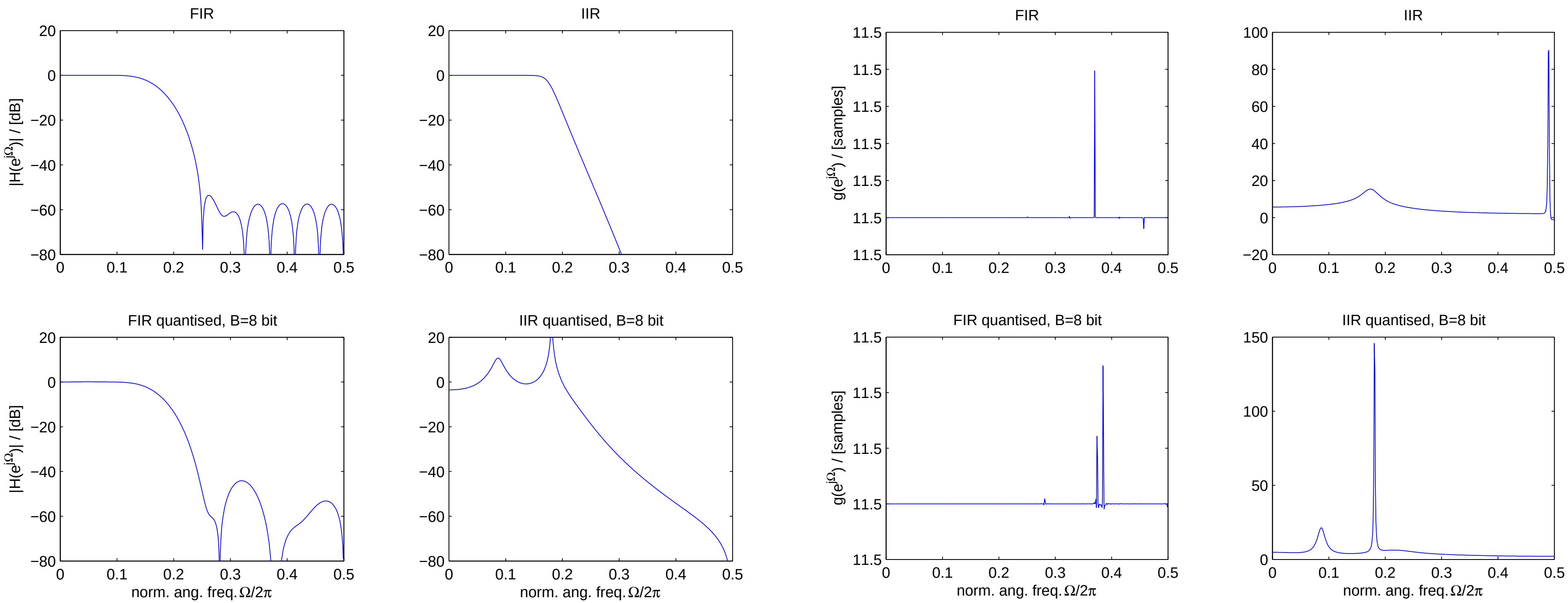
- IIR filters generally achieve high quality filters with less coefficients than FIR.

- Appreciate the response length (again geometric series):

$$\frac{1}{1 - az^{-1}} = \sum_{i=0}^{\infty} a^i \, z^{-i} \qquad \text{for} \quad |a| < 1 \tag{51}$$

FIR/IIR Comparison

- Comparison between FIR and IIR filters (both 24 coefficients):



- note that quantisation can affect an IIR filter badly (instability)!

Computational Complexity I

- Recall for general IIR filter:

$$y[n] = \sum_{l=0}^{L-1} b_l x[n-l] + \sum_{j=1}^{J-1} a_j y[n-j] = \mathbf{b}^{\mathrm{H}} \mathbf{x}_n \quad + \quad \mathbf{a}^{\mathrm{H}} \mathbf{y}_{n-1} \tag{52}$$

- whereby vector notation:

$$\mathbf{x}_n \quad = \quad [x[n] \;\; x[n-1] \quad \cdots x[n-L+1]]^{\mathrm{T}} \tag{53}$$

$$\mathbf{y}_{n-1} \quad = \quad [y[n-1] \;\; y[n-2] \quad \cdots y[n-J+1]]^{\mathrm{T}} \tag{54}$$

$$\mathbf{b} \quad = \quad [b[0] \;\; b[1] \quad \cdots b[L-1]]^{\mathrm{H}} \tag{55}$$

$$\mathbf{a} \quad = \quad [a[1] \;\; a[2] \quad \cdots a[J-1]]^{\mathrm{H}} \tag{56}$$

$$\tag{57}$$

Computational Complexity II

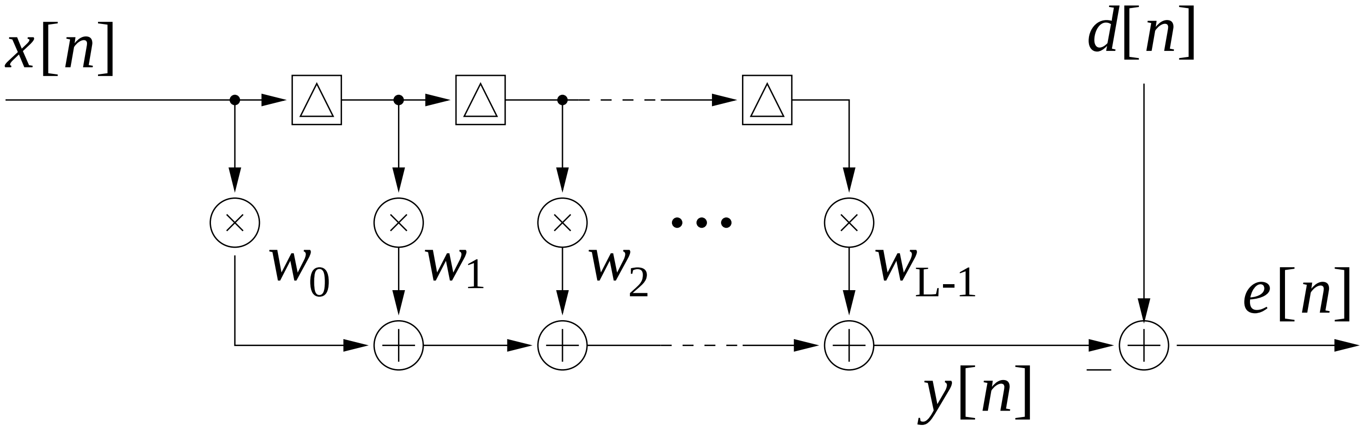
- Hence required steps per sampling period:
  - update TDL (memory moves
  - calculate $L + J$ multiply-accumulates (MAC)
- **Example 1.** Sampling rate $f_\mathrm{s} = 8$ kHz; we want to model the impulse response of an acoustic system with a $T = .5$ s duration, and therefore need $T \cdot f_\mathrm{s} = 4000$ coefficients. Total complexity: $C = 4000 f_\mathrm{s} = 32$ MMAC/s;
- **Example 2.** We double the sampling rate to wideband audio with $f_\mathrm{s} = 16$ kHz; we require 8000 coefficients and yield a total complexity of $C = 128$ MMAC/s;
- Note: doubling the sampling rate means quadrupling the complexity!
- State-of-the-art digital signal processors perform approximately 1 GMAC/s.

Summary on FIR/IIR Filtering

- Characterisation by impulse response, frequency response, or transfer function;
- Be aware of properties such as causality and minimum/maximum/non-minimum phase;
- **Stability.** FIR is strictly stable, IIR is not;
- **Quality/Complexity.** IIR filters need less coefficients than FIR.
- **Linear Phase.** For FIR filters a symmetric impulse response is necessary and sufficient for linear phase; IIR are never linear phase (although some yield an approximation in the passband);
- **Implementation.** Both FIR and IIR are sensitive to fixed point implementation, but IIR has larger dynamics in its filter coefficients and may even become unstable when quantised.

Adaptive Filtering

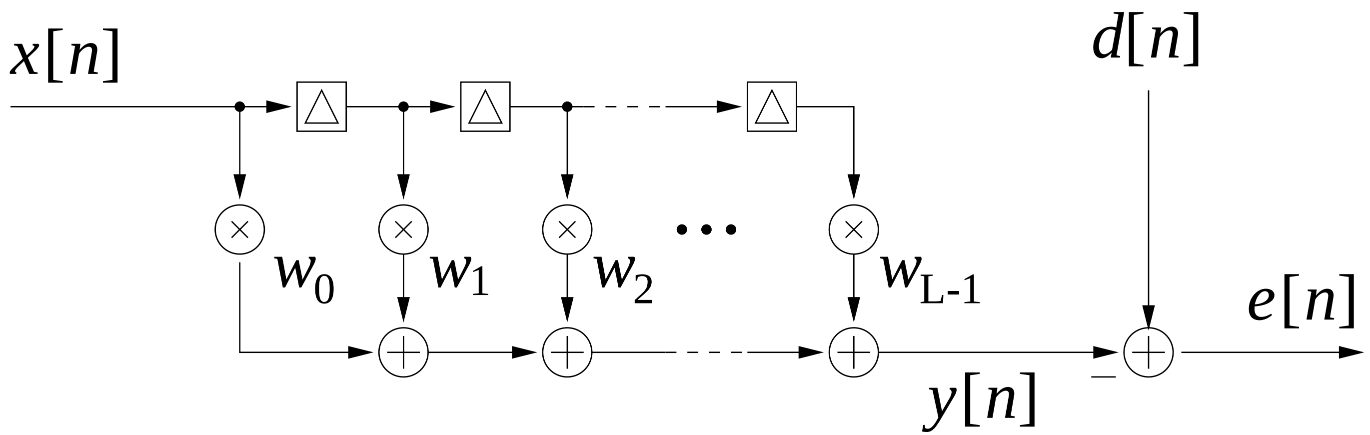
- Previously we have designed filters with specific characteristics; reconsider an FIR filter with L coefficients $w[n]$:



- here the adjustment of the coefficients $w[n]$ may be dependent on the environment, and the coefficients may be variable over time.

Generic Adaptive Filtering

- The architecture of a generic adaptive filter comprises of an input signal $x[n]$, a desired signal $d[n]$, a filter output $y[n]$, an error signal $e[n]$, and the filter impuls response $w[n]$;

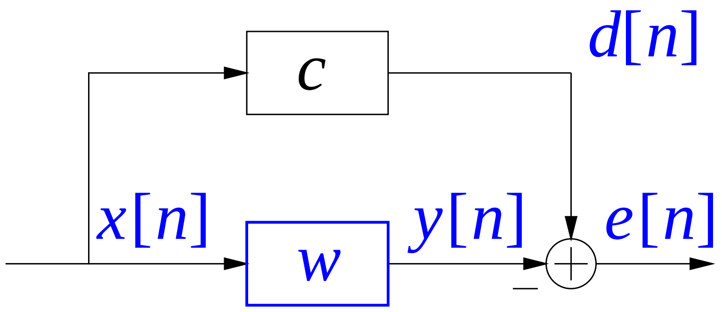


- **Aim of an adaptive filter:** the filter $w[n]$ forms an output $y[n]$ from an input $x[n]$, such that, when subtracted from the desired signal $d[n]$, the reslting error is minimised in a suitable sense.

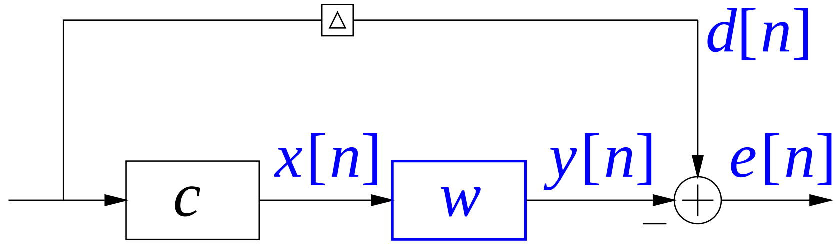
Adaptive Filter Architectures

- A generic adaptive filter can be applied in different architectures:

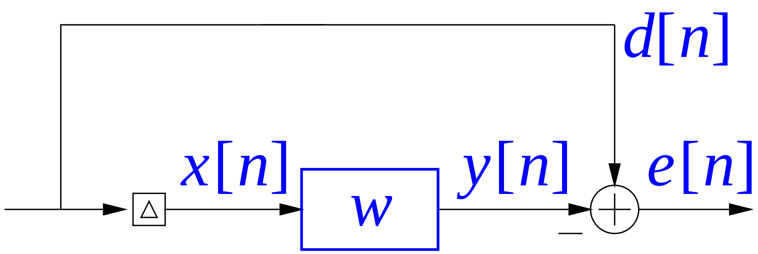
- system identification



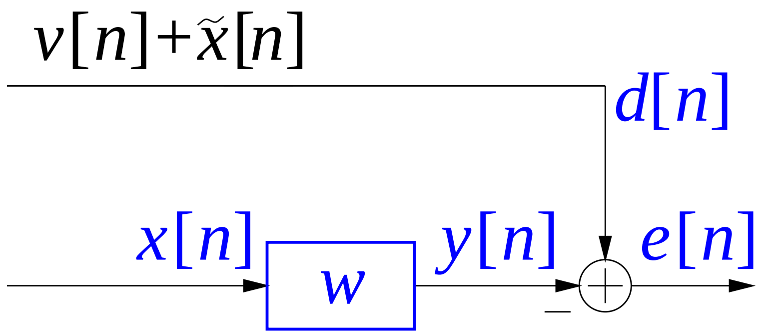
- inverse system identification / equalisation



- prediction

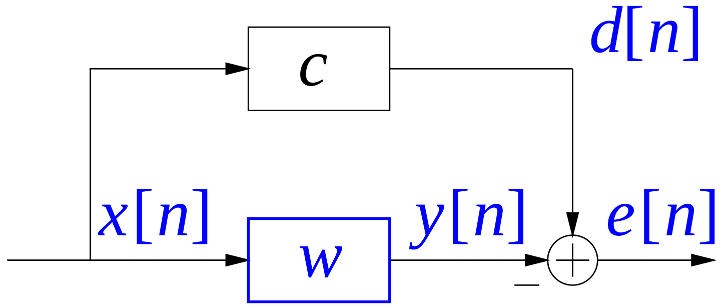


- noise cancellation



System Identification

- We want to identify an unknown LTI system with impulse response $c[n]$:

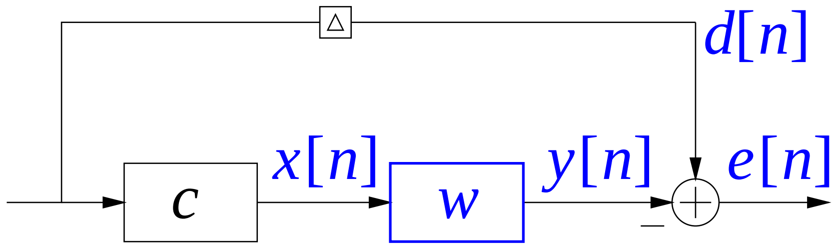


- if the adaptive filter converges and $e[n]$ is (ideally) zero, then the input-output behaviour of $c[n]$ and $w[n]$ is identical;

- if $x[n]$ is sufficiently broadband, we have identified $c[n]$ by means of a suitable model contained in $w[n]$.

Inverse System Identification / Equalisation

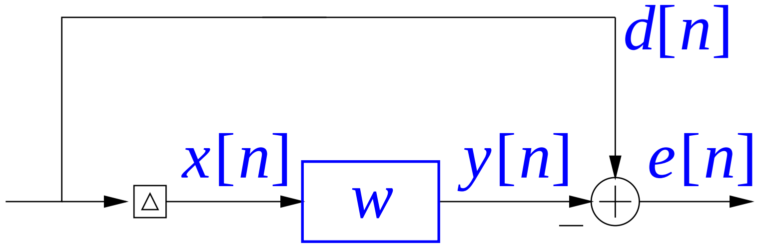
- We place the unknown system and the adaptive filter in series:



- if the error signal is minised, then $w[n]$ represents the inverse system of $c[n]$;
- if $c[n]$ is non-minimum phase, we need to permit a delay Δ, such that $c[n] * w[n] \approx \delta[n - \Delta]$.

Prediction

- Due to delaying the input signal $x[n]$, the desired signal $d[n]$ is a "future version" of the filter input:

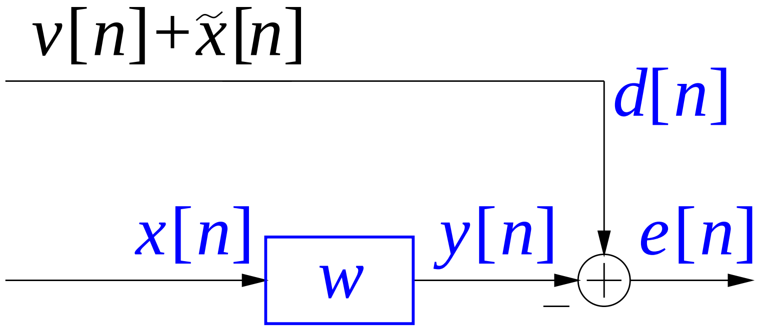


- the adaptive filter should only be able to predict "periodic" or correlated components with $d[n]$;

- the error signal $e[n]$ would in this case contain any remaining uncorrelated signal components.

Adaptive Noise Cancellation

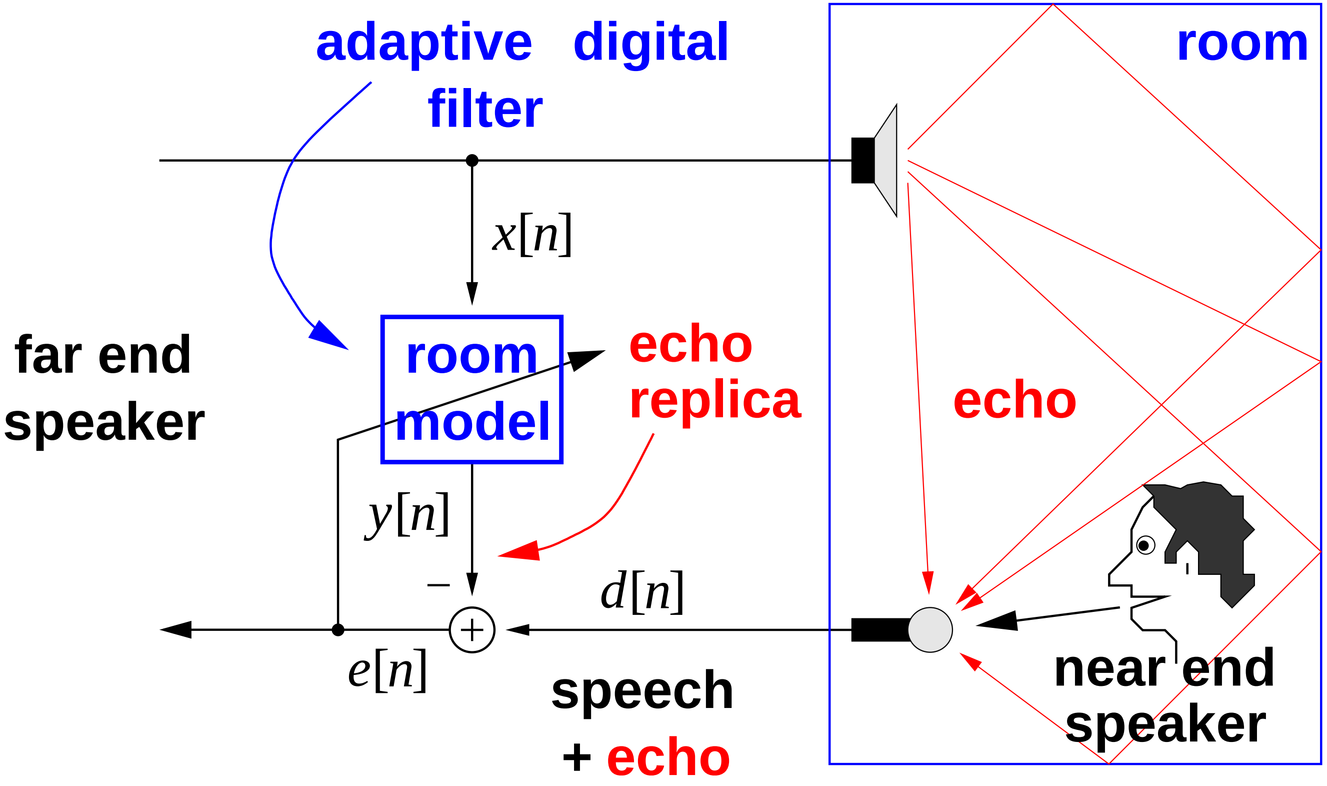
- consider a noise corrupted signal $d[n]$ (e.g. helicopter pilot's microphone signal):



- if we can obtain a reference signa $x[n]$ of the noise, the adaptive filter should be able to cancel the noise component from $e[n]$, provided that the noise in $x[n]$ and $d[n]$ is linearly related.

Acoustic Echo Cancellation

- **hands-free telephony** — full-duplex mode suffers from acoustic feedback:



  - acoustic feedback is very disturbing for the far end speaker
  - by identifying the room acoustics (over the frequency range of speech), the adaptive filter can produce a replica of the echo;

- due to the reverberation times of a room, which the adaptive filter has to match, this is a very computationally demanding task.

How Can We Optimise the Filter Coefficients?

- In order to calulate an optimal filter $w_{\text{opt}}[n]$, the mean square error (MSE) $\mathcal{E}\{e^2[n]\}$ is a good criterion to be minimised;
- Formulation of error signal $e[n]$:

$$e[n] = d[n] - y[n] = d[n] - \sum_{i=0}^{L-1} w_i \cdot x[n-i] = d[n] - \mathbf{w}^{\mathrm{H}} \cdot \mathbf{x}_n$$

- with definitions:

$$\mathbf{w} = \begin{bmatrix} w_0 & w_1 & \cdots & w_{L-1} \end{bmatrix}^{\mathrm{T}};$$
$$\mathbf{x}_n = \begin{bmatrix} x[n] & x[n-1] & \cdots & x[n-L+1] \end{bmatrix}^{\mathrm{T}};$$

- (we could do the derivation for complex valued signals, but for simplicity here stick with the real valued case.)

Optimal Filter — MSE Formulation

- Mean Square Error formulation (note that the optimum filter $\mathbf{w}_{\text{opt}}$ is deterministic!):

$$\begin{aligned}\mathcal{E}\{e^2[n]\} &= \mathcal{E}\{(d[n] - \mathbf{w}^{\mathrm{T}} \cdot \mathbf{x}_n)^2\} \\ &= \mathcal{E}\{d^2[n]\} - 2\mathbf{w}^{\mathrm{T}} \cdot \mathcal{E}\{d[n] \cdot \mathbf{x}_n\} + \mathbf{w}^{\mathrm{T}} \cdot \mathcal{E}\{\mathbf{x}_n \cdot \mathbf{x}_n^{\mathrm{T}}\} \cdot \mathbf{w} \\ &= \sigma_d^2 \;-\; 2\mathbf{w}^{\mathrm{T}} \cdot \mathbf{p} \;+\; \mathbf{w}^{\mathrm{T}} \cdot \mathbf{R} \cdot \mathbf{w}\end{aligned}$$

  with covariance matrix $\mathbf{R} = \mathcal{E}\{\mathbf{x}_n \cdot \mathbf{x}_n^{\mathrm{T}}\}$ and correlation vector $\mathbf{p} = \mathcal{E}\{d[n] \cdot \mathbf{x}_n\}$;

- note: $\mathbf{w}^{\mathrm{T}} \cdot \mathbf{x}_n$ is a scalar, hence $\mathbf{w}^{\mathrm{T}} \cdot \mathbf{x}_n = (\mathbf{w}^{\mathrm{T}} \cdot \mathbf{x}_n)^{\mathrm{T}} = \mathbf{x}_n^{\mathrm{T}} \cdot \mathbf{w}$;

- important: the expression for $\mathcal{E}\{e^2[n]\}$ is quadratic in the filter coefficients $\mathbf{w}$; hence optimisation should lead to a unique extremum!

Covariance Matrix & Cross-Correlation Vector

- Given the auto-correlation sequence $r_{xx}[\tau]$ and the cross-correlation sequence $r_{xd}[\tau]$:

$$r_{xx}[\tau] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]\cdot x[n-\tau] \qquad\qquad r_{xd}[\tau] = \frac{1}{N}\sum_{n=0}^{N-1} d[n]\cdot x[n-\tau]$$

- the covariance matrix $\mathbf{R}$ and cross-correlation vector $\mathbf{p}$ can be written

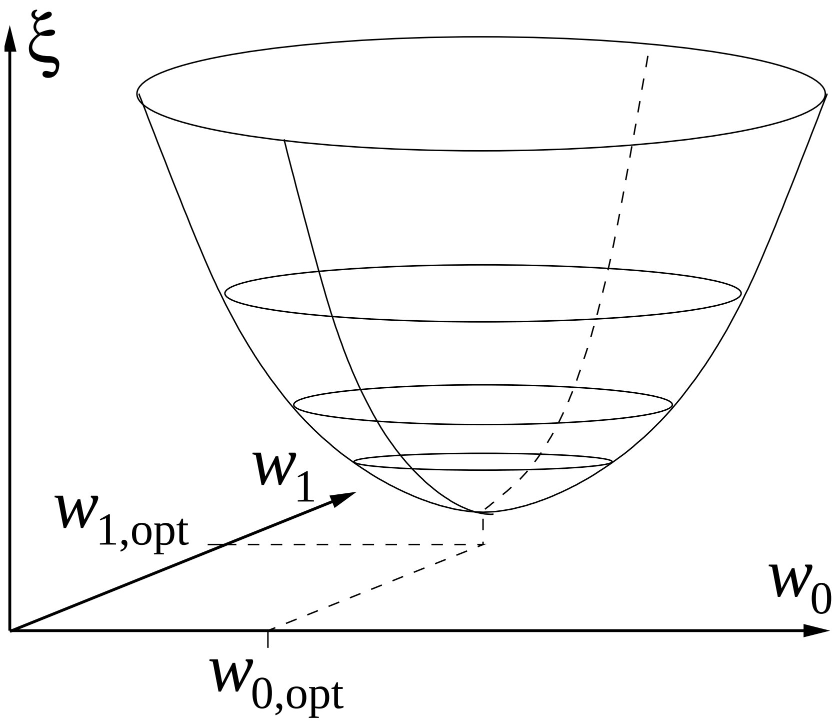
$$\mathbf{R} = \left[\begin{array}{cccc} r_{xx}[0] & r_{xx}[1] & \cdot & r_{xx}[L-1] \\ r_{xx}[1] & r_{xx}[0] & \cdot & r_{xx}[L-2] \\ \vdots & & \ddots & \vdots \\ r_{xx}[L-1] & r_{xx}[L-2] & \cdots & r_{xx}[0] \end{array}\right] \qquad \mathbf{p} = \left[\begin{array}{c} r_{xd}[0] \\ r_{xd}[1] \\ \vdots \\ r_{xd}[L-1] \end{array}\right]$$

- the Wiener-Hopf solution requires estimation of $\mathbf{R}$ and $\mathbf{p}$, and a matrix inversion (numerically costly and potentially unstable!).

MSE Surface

- Quadratic in the filter coefficients $\mathbf{w}$, the surface of the MSE $\xi = \mathcal{E}\{e^2[n]\}$ is a hyperparabola in $L+1$ dimensional hyperspace;

- example for $L = 2$ coefficients:



Optimum Filter — Minimum Mean Squared Error

- Standard optimisation procedure to achieve minimum MSE:

$$\frac{\partial}{\partial \mathbf{w}} \mathcal{E}\left\{e^2[n]\right\} \stackrel{!}{=} \mathbf{0}$$

- Inserting the previous expression for the MSE:

$$\frac{\partial}{\partial \mathbf{w}} \mathcal{E}\left\{e^2[n]\right\} = \mathbf{0} - 2\mathbf{p} + 2\mathbf{R} \cdot \mathbf{w}$$

- if $\mathbf{R}$ is invertible, then the optimum filter coefficients $\mathbf{w}_{\text{opt}}$ are given by (Wiener-Hopf equation):

$$\mathbf{w}_{\text{opt}} = \mathbf{R}^{-1} \cdot \mathbf{p}$$

Optimum Filter — Minimum Mean Squared Error

- Standard optimisation procedure to achieve minimum MSE:

$$\frac{\partial}{\partial \mathbf{w}} \mathcal{E}\left\{e^2[n]\right\} \stackrel{!}{=} \mathbf{0}$$

- Inserting the previous expression for the MSE:

$$\frac{\partial}{\partial \mathbf{w}} \mathcal{E}\left\{e^2[n]\right\} = \mathbf{0} - 2\mathbf{p} + 2\mathbf{R} \cdot \mathbf{w}$$

- if $\mathbf{R}$ is invertible, then the optimum filter coefficients $\mathbf{w}_{\text{opt}}$ are given by (Wiener-Hopf equation):

$$\mathbf{w}_{\text{opt}} = \mathbf{R}^{-1} \cdot \mathbf{p}$$

Viability of Wiener-Hopf

- Confident estimation of $\mathbf{R}$ and $\mathbf{p}$ requires long time windows;
- inversion of $\mathbf{R}$ is of order $\mathcal{O}(L^3)$ — very complex!
- Wiener-Hopf requires $\mathbf{R}$ to be invertible;
- even if $\mathbf{R}$ has full rank, it may be ill-conditioned.

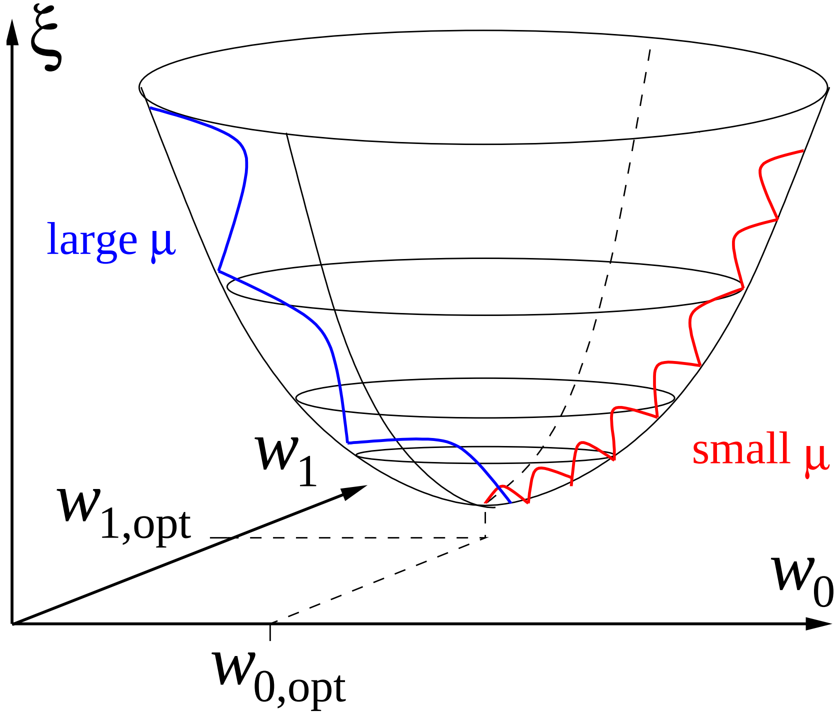
Iterative Solutions

- In order to bypass the problems of Wiener Hopf, we will utilise iterative approaches to converge towards the optimal solution;

  - Starting from an initial guess $\mathbf{w}[0]$, gradient descent techniques follow the negative gradient of the MSE cost function:

$$\mathbf{w}[n+1] = \mathbf{w}[n] - \mu\nabla\xi[n]$$

  - μ is called the "step size" of the algorithm and controls convergence . . .



Gradient Descent Techniques

- We require the gradient $\nabla\xi[n] = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}[n]$ — see slide 81
- therefore:
$$\mathbf{w}[n+1] = \mathbf{w}[n] + 2\mu(\mathbf{p} - \mathbf{R}\mathbf{w}[n]) \quad (58)$$
- this is also know as the method of steepest descent;
- note I: matrix inversion is no longer required;
- note II: we still need to estimate $\mathbf{R}$ and $\mathbf{p}$.

Stochastic Gradient Techniques

- We omit expectations and minimise the instantaneous squared error:

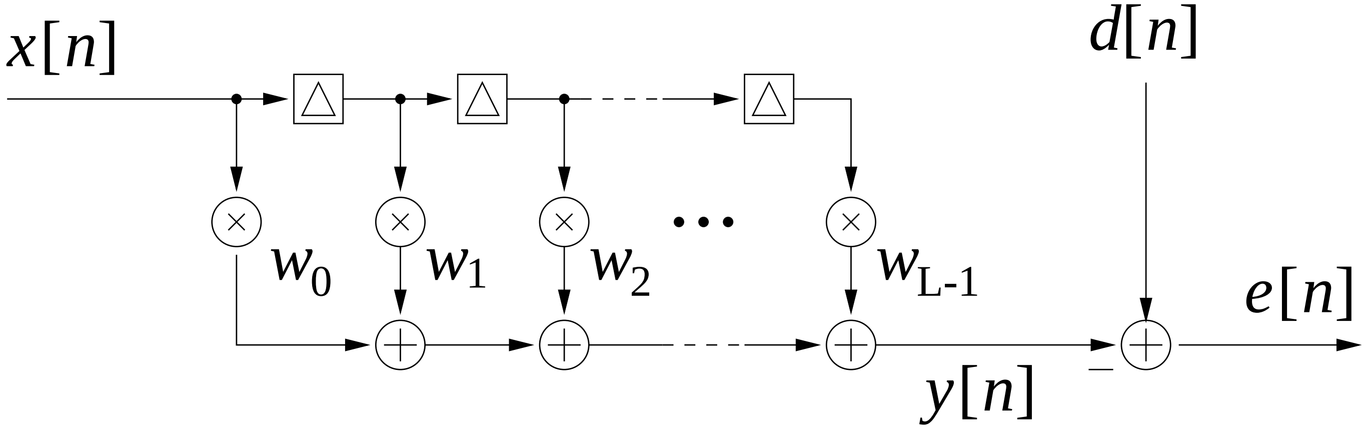
$$\mathbf{w}e^2[n] = -2e[n] \cdot \mathbf{x}_n \quad (59)$$

- this results in the least mean squares (LMS) algorithm:

$$\mathbf{w}[n+1] = \mathbf{w}[n] + 2\mu e[n] \cdot \mathbf{x}_n \quad (60)$$

- the gradient estimate is noise, but through itertion, *on average* the gradient estimate points into the right direction.

Least Mean Squares Algorithm



- Steps for the LMS adaptive filter:
  (1) update tap delay line vector $\mathbf{x}[n]$
  (2) calculate filter output: $y[n] = \mathbf{w}^{\mathrm{H}}[n]\mathbf{x}[n]$
  (3) calculate error: $e[n] = d[n] - y[n]$
  (4) LMS update: $\mathbf{w}[n+1] = \mathbf{w}[n] + 2\mu e[n] \cdot \mathbf{x}_n$

- Complexity: $2L$ multiply accumulate (L for scalar product, L for LMS update) per sampling period.

LMS Convergence

- The step size μ controls the convergence of the LMS;
- intuitive from MSE surface plot: small (large) $\mu \longrightarrow$ slow (fast) convergence with high (low) final accuracy;
- Without proof: convergence can be guaranteed if

$$0 \ < \ \mu \ < \ \frac{1}{L\sigma_{xx}^2} \tag{61}$$

where L is the filter length and σ_{xx}^2 the variance of the input signal.

LMS System Identification Example I

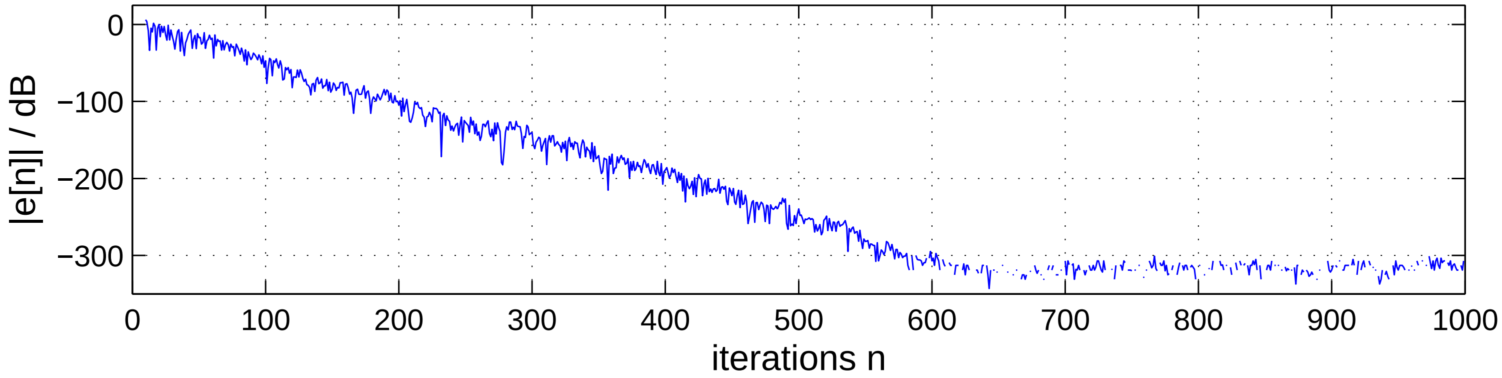
- Matlab code for LMS algorithm in system identification:

```
x = randn(1,1000);                    % input signal
c = [1 0 .5 -.3];                     % ‘‘unknown’’ system
d = filter(c,1,x);                    % desired signal
L = 10;                               % filter length
mu = 0.01;                            % step size

w = zeros(L,1);                       % weight vector initialisation
for n = L:1000,
    Xtdl = x(n:-1:n-L+1)’;            % update TDL
    y(n) = w’*Xtdl;                   % calculate filter output
    e(n) = d(n)-y(n);                 % calculate error
    w = w + 2*mu*e(n)*Xtdl;           % LMS weight update
end;
```

LMS System Identification Example II

- Convergence of error:



- The error reaches Matlab's floating point accuracy of approximately 10^{-16} — this is numerically zero.

- after convergence, the filter coefficients match the impulse response of the unknown system $c[n]$.

Channel Equalisation Example I

- Assume a minimum phase channel $C(z) = 1 + 0.5z^{-1}$;
- the inverse of this channel is given by

$$W(z) = \frac{1}{C(z)} = \frac{1}{1 + 0.5z^{-1}} \tag{62}$$

- we can use the geometric series, $\frac{1}{1-a} = \sum_{n=0}^{\infty} a^n$ for $|a| < 1$, to find an FIR approximation of the inverse:

$$W(z) = \sum_{n=0}^{\infty} (-.5)^n z^{-n} = 1 - .5z^{-1} + .25z^{-2} - 0.125z^{-3} + \cdots \tag{63}$$

- For uncorrelated input, an adaptive filter of length L would identify the first L elements of this geometric series.

Channel Equalisation Example II

- Now assume a maximum phase channel $C(z) = .5 + z^{-1}$;
- the inverse of this channel is unstable

$$W(z) = \frac{1}{C(z)} = \frac{2}{1+2z^{-1}} = 2 - 4z^{-1} + 8z^{-2} - \cdots \quad (64)$$

and the "geometric series" diverges; alternatively

$$W(z) = \frac{z}{1+0.5z} = z\sum_{n=0}^{\infty}(-.5)^n z^n = z - .5z^1 + .25z^2 - \cdots \quad (65)$$

converges but is anti-causal.

Channel Equalisation Example II ctd.

- If we permit a delay of e.g. 3 samples, z^{-3}, such that

$$W(z) = \frac{z^{-2}}{1+0.5z} = z^{-2}\sum_{n=0}^{\infty}(-.5)^n z \quad (66)$$

$$= z^{-2} - 0.5z^{-1} + 0.25 - 0.125z^{1} + \cdots \quad (67)$$

an adaptive filter can identify the 3 causal coefficients of the geometric series.

- hence, for equalisation or inverse system identification, in general we have to permit a delay for the system $C(z)W(z)$;

- rule of thumb: not knowing anything about thye phase properties of $C(z)$, practisioners choose a delay of $L/2$.

Channel Equalisation Example III

- Now consider a channel $C(z) = 1 - z^{-1}$ — this is a highpass filter with a zero at $z = 1$ cancelling the dc ($f = 0$) frequency component;

- the inverse system would attempted to provide a very high (ideally infinite) gain at $f_s/2$ — this is very dangerous!

- inspect the covariance matrix $\mathbf{R}$; for uncorrelated input $u[n]$ to the channel, the autocorrelation function of $x[n]$ is given by

$$r_{xx}[\tau] = c[\tau] * c[-\tau] \tag{68}$$

  and is likely to be ill-conditioned.

- try to establish what filter response the LMS algorithm converges to.

Channel Equalisation Example III ctd.

- In a realistic scenario, a channel suffers from additive white Gaussia noise $v[n]$, such that

$$x[n] = c[n] * u[n] \quad + \quad v[n] \tag{69}$$

whereby $v[n]$ is independent of $u[n]$;

- therefore for the autocorrelation function due to independence:

$$r_{xx}[\tau] = c[\tau] * c[-\tau] \quad + \quad r_{vv}[\tau] \tag{70}$$

- for the covariance matrix $\mathbf{R}$, we obtain

$$\mathbf{R}_{xx} = \mathbf{R}_{cc} \quad + \quad R_{vv} \tag{71}$$

whereby $\mathbf{R}_{cc}$ is the covariance matrix of the noiseless channel output, and $R_{vv} = \sigma_{vv}^2 \mathbf{I}$ the covariance matrix of the AWGN.

Channel Equalisation Example III ctd.

- Comparing $\mathbf{R}_{cc}$ and $\mathbf{R}_{xx}$, the conditioning of the matrix is improved (

$$\mathbf{R}_{xx} = \underbrace{\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^{\mathrm{H}}}_{\mathbf{R}_{cc}} + \sigma_{vv}^2 \mathbf{I} = \mathbf{Q} \begin{bmatrix} \lambda_{c,0} + \sigma_{vv}^2 & 0 & & 0 \\ 0 & \lambda_{c,1} + \sigma_{vv}^2 & 0 & 0 \\ \vdots & & \ddots & \vdots \\ 0 & & 0 & \lambda_{c,L-1} + \sigma_{vv}^2 \end{bmatrix} \mathbf{Q}^{\mathrm{H}} \tag{72}$$

- simulate the previous example with channel noise: in the presence of channel noise, the adaptive filter strikes a balance between inversion of $C(z)$ and noise amplification;

- the adaptive filter provides a minimum mean square error (MMSE) solution rather than an inversion: the system inverse is "regularised" by the AWGN.

Summary / Further Study

- Adaptive filters tuneable filter coefficients and can be applied in situation where exactly desired response (highpass, lowpass, etc) is not known a priori;
- many other popular adaptive algorithms exist beyond the LMS;
- for further reading:

[1] B. Widrow and S.D. Stearns *Adaptive Signal Processing*. Prentice Hall, 1985. *[Classical and easy to read text.]*

[2] S. Haykin *Adaptive Filter Theory*. Prentice Hall, 1987, 1991, and 1996. *[A good mathematical treatment — strangely the 2nd edition is considered better than the third.]*

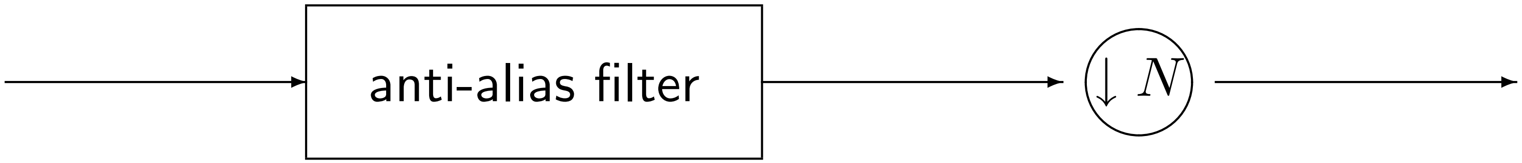
7. Multirate Signal Processing

- In a multirate system, various components run at different sampling rates;
- Audio DSP: you may need to convert from professional audio (96kHz) down to HIFI (e.g. digital audio tape, 48kHz) or “wideband” audio (teleconferencing, 16kHz);
- Software defined radio (SDR): you may need to create a digital IF signal (sampled at, say, 75MHz) from a UMTS baseband signal (5MHz sampling rate);

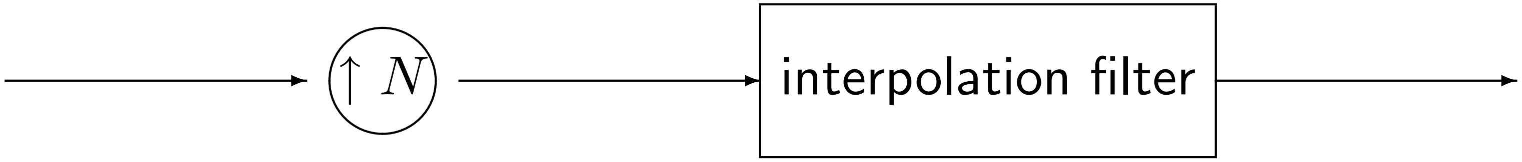
  Similarly, in an SDR receiver, you may have to extract a symbol stream at 5MHz from a 75MHz IF signal.

Multirate Operations

- A multirate system usually comprises of digital filters, decimators, and expanders;
- anti-alias filtering followed by N-fold decimation (downsampling) to lower the sampling rate by a factor N:

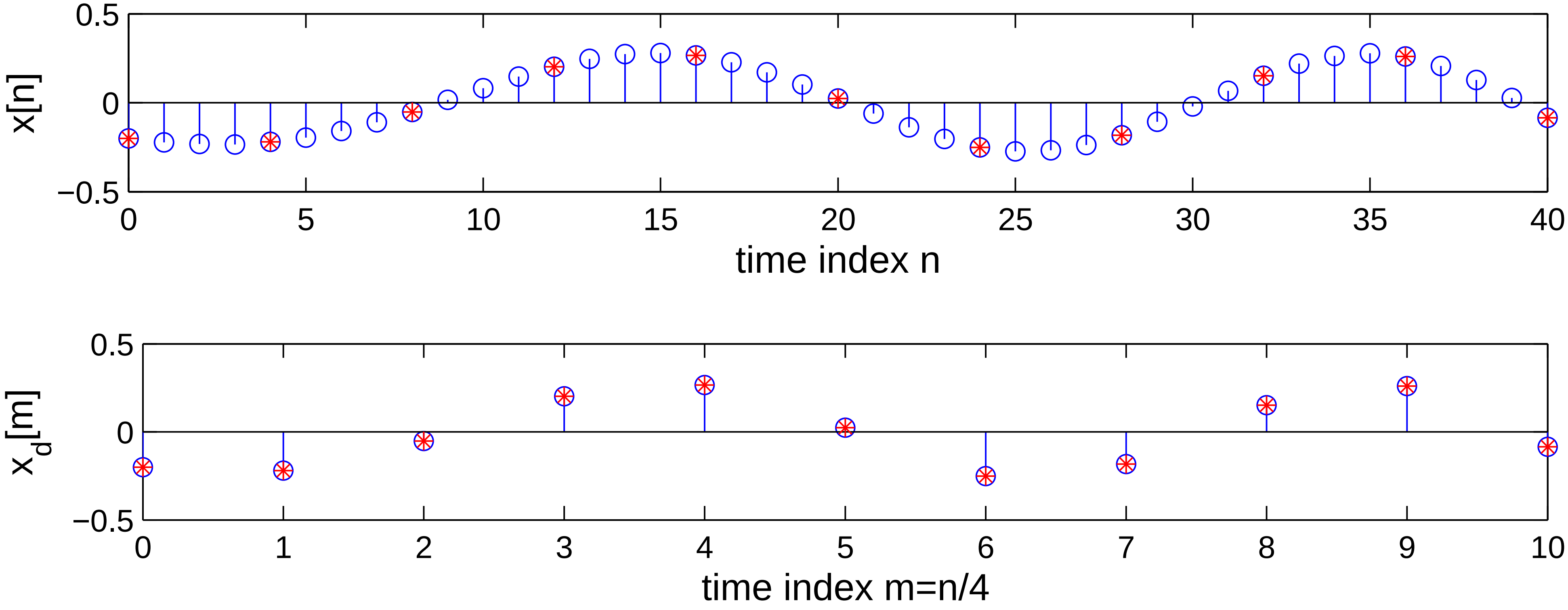


- N-fold expansion or up-sampling followed by interpolation filtering to increase the sampling rate by a factor of N:



Decimation — Time Domain Desciption

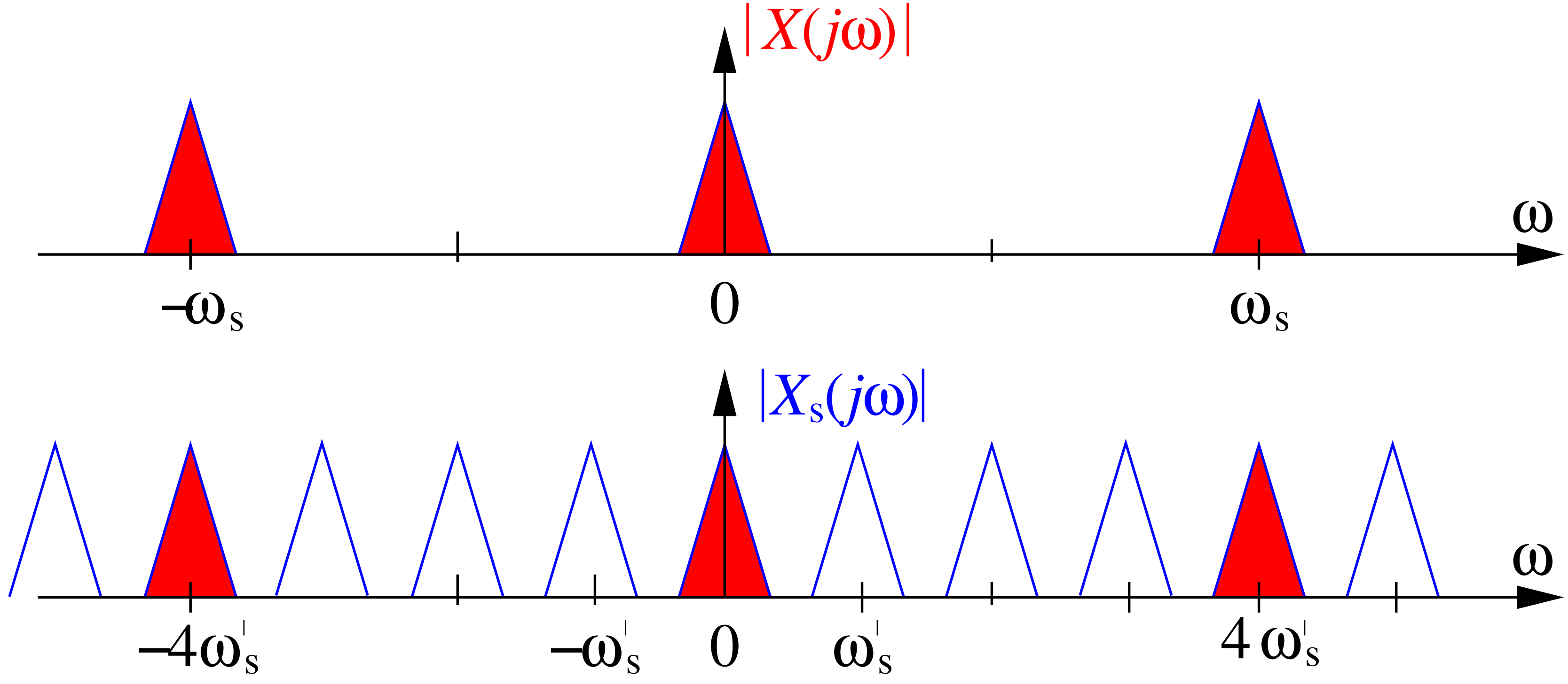
- To lower the sampling rate by a factor of N, a decimator or downsampler only retains every Nth sample — other samples are discarded; exampled for $N = 4$:



- if the signal is suitably bandlimited, we loose no information in the original signal; also, the exact sampling point can be chosen arbitrarily.

Decimation — Frequency Domain Description

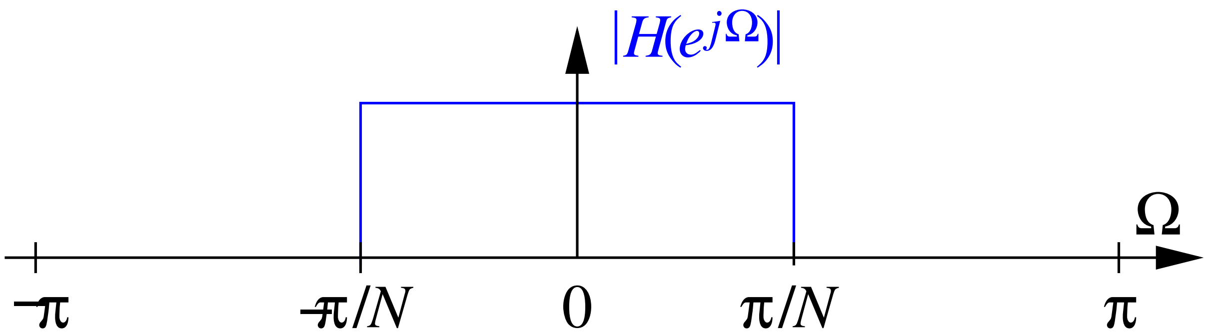
- similar to sampling on slide 13, resampling "periodises" the spectrum with respect to the new sampling rate $\omega_s' = \omega_s/N$; example for $N = 4$ (with unnormalised angular frequencies):



- note: if the original signal is not suitably bandlimited, aliasing may occur.

Anti-Alias Filtering

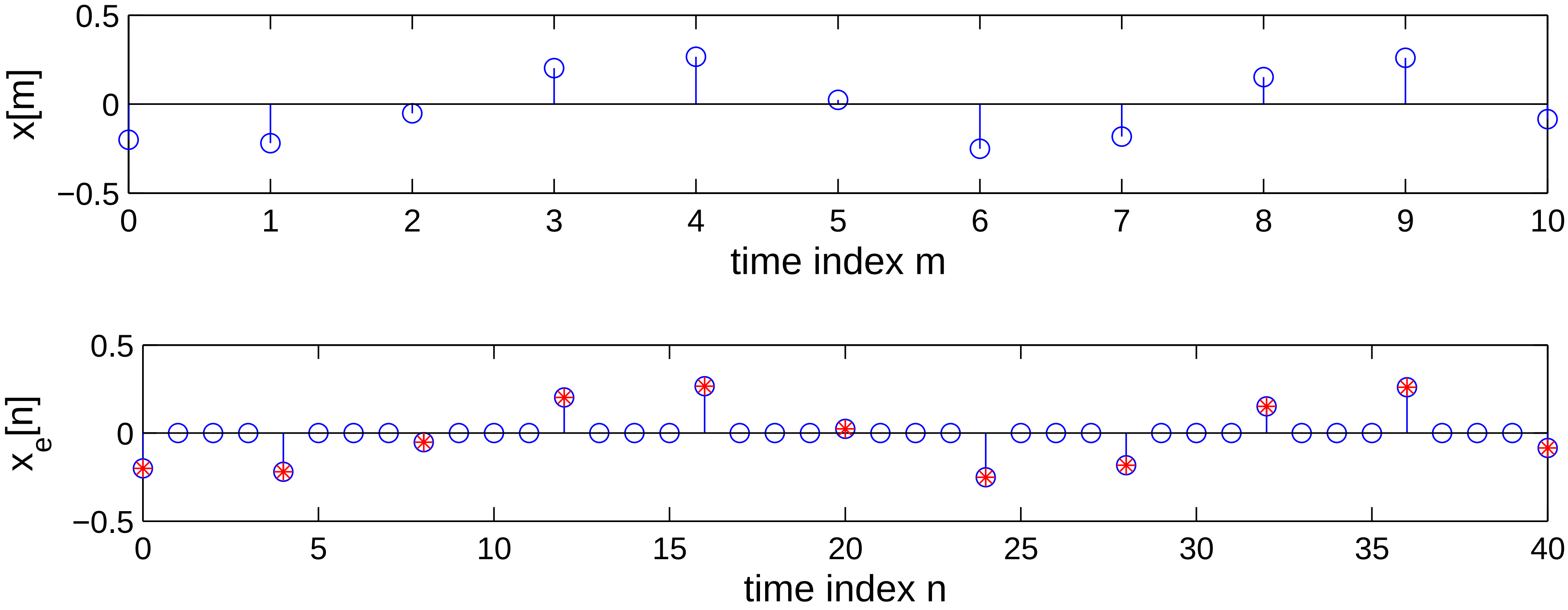
- A digital anti-alias filter can be employed to filter the signal prior to decimation:



- it is difficult to create a brick-wall type filter; however, in the digital domain we can achieve a high quality filter design at a lower "cost" than in the analogue domain.

Expansion — Time Domain Description

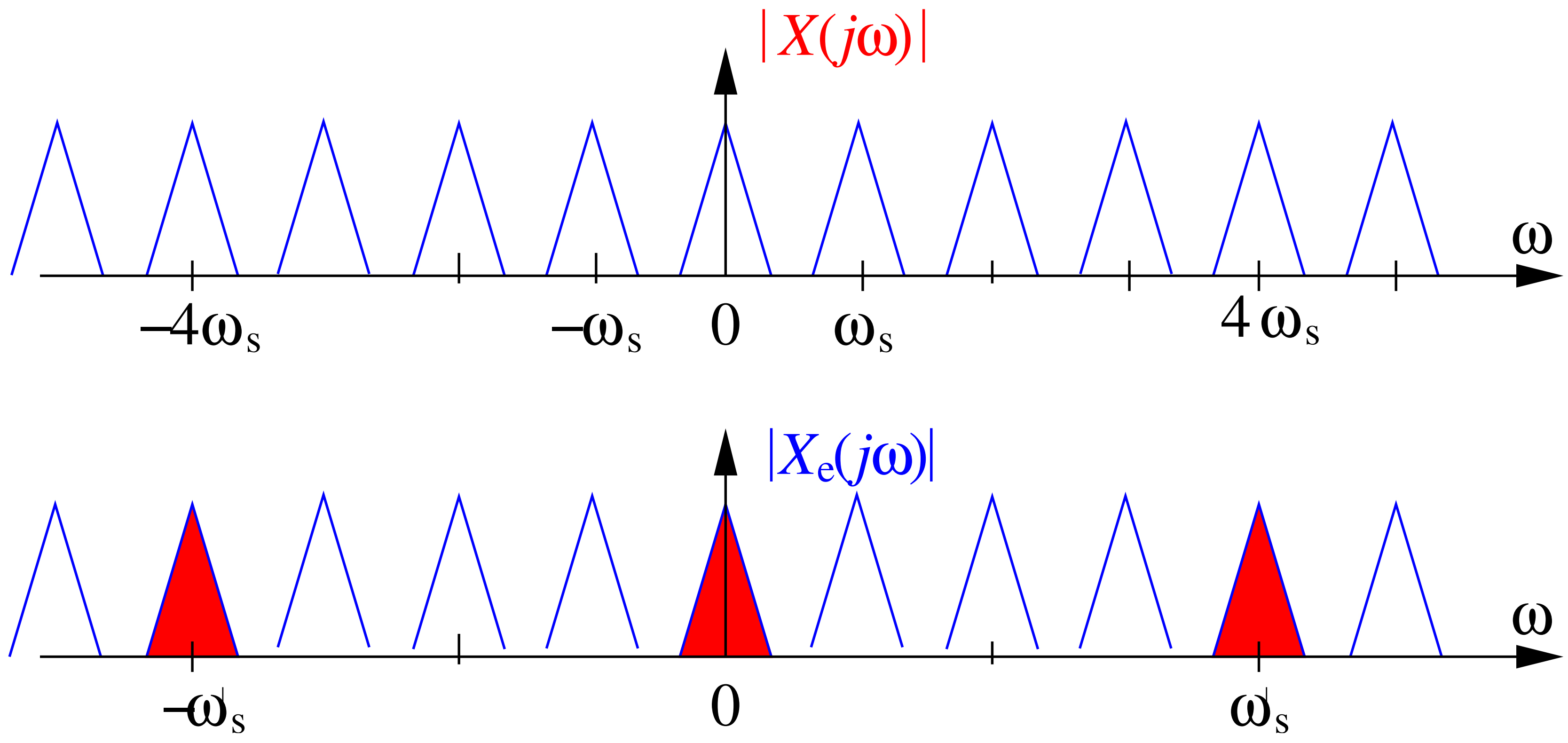
- To increase the sampling rate by a factor of N, an expander or upsampler fill $N-1$ zero sample in between every other sample; exampled for $N=4$:



- note that our signal is now impulsive . . . our signal can be expected to contain some high frequency components.

Expansion — Frequency Domain Description

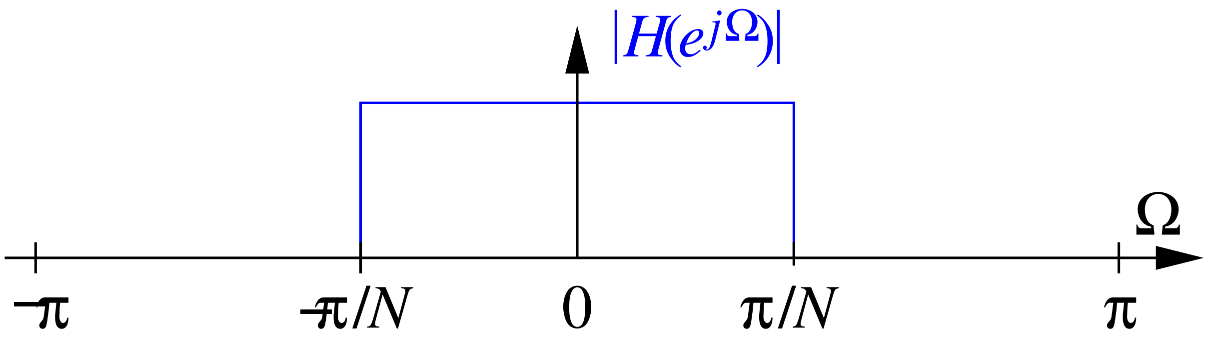
- Expansion simply re-scales the frequency axis:



- in the expanded signal, we see $N-1$ spectral repetions in addition to the original (baseband) signal;

Expansion — Interpolation Filter

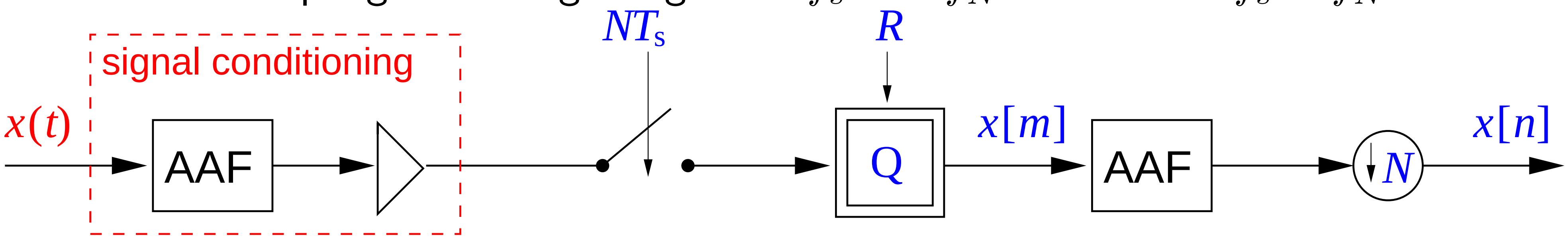
- A digital interpolation or reconstruction filter can be employed to retireve the baseband portion of the signal after expansion:



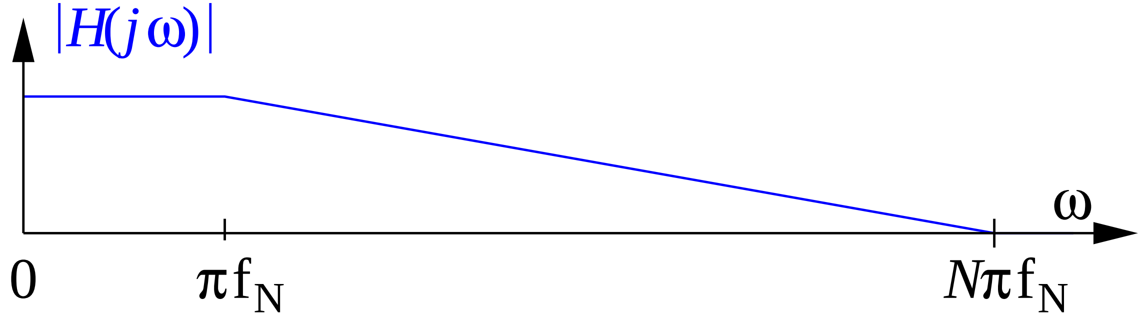
- this is identical to the conditions required of the anti-alias filter.

Oversampled ADC

- Consider sampling an analogue signal at $f_s = N f_N$ rather than $f_s = f_N$:



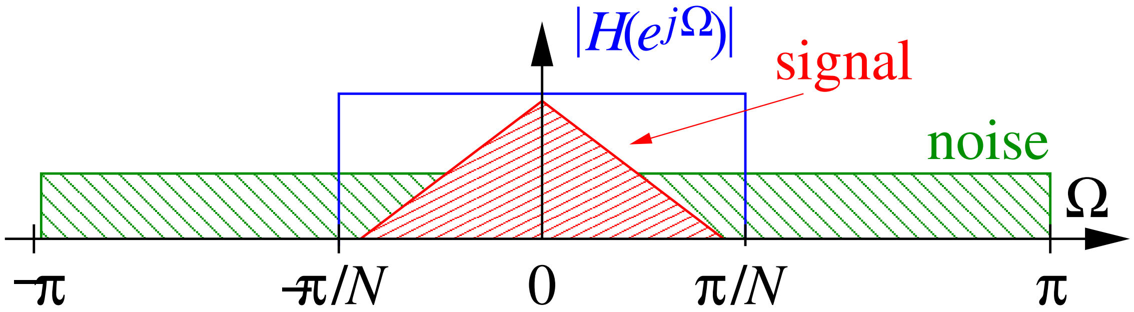
- the analogue anti-alias filter can now be relaxed:



- the transition band of the anti-alias filter can be much wider than in the non-oversampled case; to compensate, we will postprocess the oversampled digital $x[m]$ by (much cheaper) digital operations.

Digital Postprocessing in Oversampled ADCs

- Requirement for the digital anti-alias filter is a tight cut-off at $\Omega_c = \pi/N$:



- this is easier to achieve than a high quality analogue filter;

- additionally, the digital AAF reduces the quantisation noise power $(q^2/12)$ by a factor $1/N$ (recall that the area under PSD is a measure for the power!).

Virtual Increase in Word Length

- Assume that the original signal was quantised with a word length R;
- the quantisation noise power after the digital anti-alias filter has removed the out-of-band noise for $N = 4$:

$$\sigma_{\mathrm{q}}^2 = \frac{1}{4}\,\frac{q^2}{12} = \frac{(q/2)^2}{12} \tag{73}$$

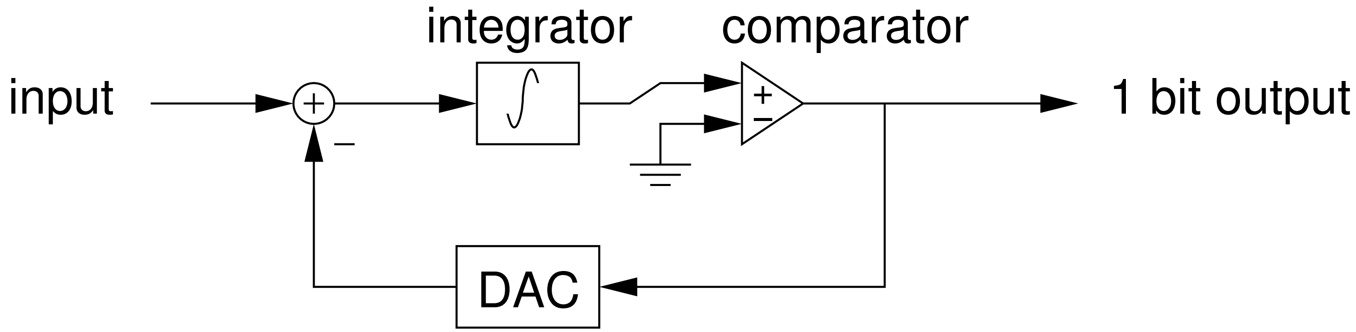
- if the quantisation step size is halved, we have gained an extra bit in resolution;
- for every quadrupling of the oversampling ratio N, we achieve an extra bit in resolution.

Trade-off between Sampling and Wordlength

- We can oversample our analogue signal by $4^{\Delta R}$ to achieve ΔR extra bits in resolution after digital postprocessing;
- to achieve 16 bit quality from a single bit quantiser, we would have to oversampled by a factor $4^1 5$;
- single bit converters exist, but use an additional technique, whereby the quantisation noise is "pushed" towards higher frequencies and can be better suppressed by a digital lowpass anti-alias filter.

Sigma-Delta Converter

- A sigma-delta converter is a mixed analogue-digital device, which provides a low-bit (often 1 bit) output:



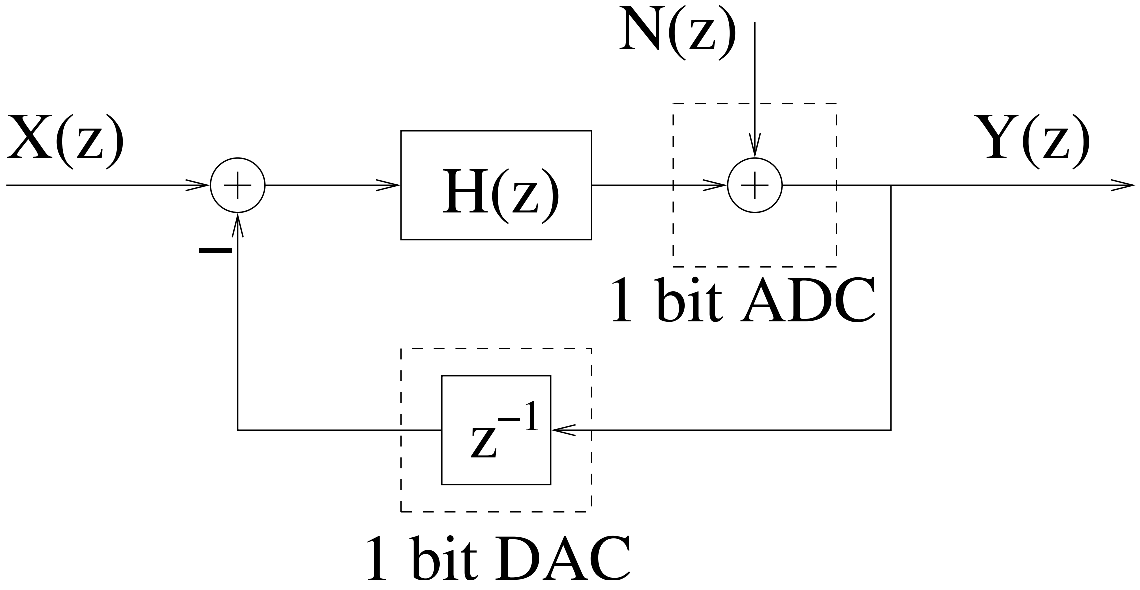
- The input sigma is passed through an integrator (lowpass filter), and the output is converted with single bit accuracy, i.e. only retaining the sign;

- the output bit-stream is converted into a voltage signal by a DAC and subtracted from the input.

-

Discrete Time Model

- In order to analyse the sigma-delta circuit, we approximate it by a discrete time representation, with input $X(z)$, white additive noise $N(z)$, and output $Y(z)$:

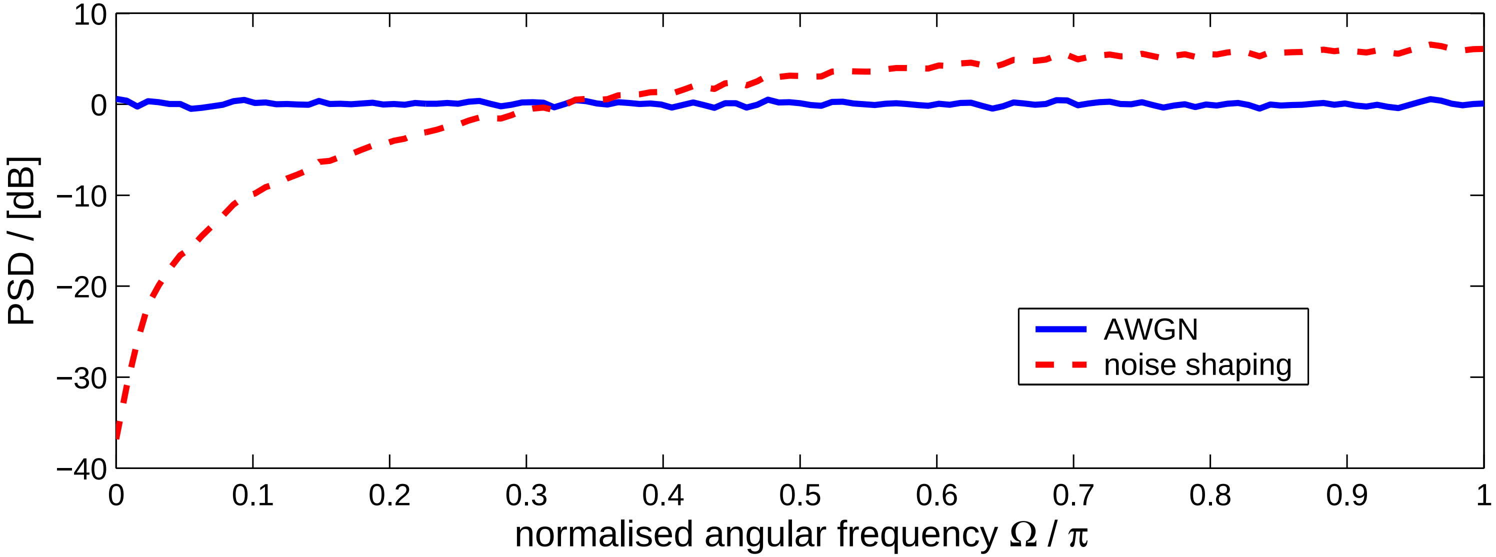


- it is easy to verify that $Y(z) = X(z) + (1 - z^{-1})N(z)$;

- this means that the output contains the desired signal plus noise with a highpass PSD.

Extracting High Bit Resolution

- The quantisation noise before and after noise shaping:



(PSD of $N(z)$ is solid blue, noise component in $Y(z)$ is dashed red)

- if we apply this in combination with oversampling, due to the reduced noise power in the baseband, substantial resolution can be gained.

Summary

- We have considered multirate signal processing components, which permit to change the sampling rate;
- oversampling ADCs have been considered as an application; although not considered here, oversampling can be similarly exploited to reduce the cost and increase the performance of DACs;
- Additional benefit can be gained by noise shaping in combination with oversampling; ultimately, a singe bit converter can provide multi-bit quality (assuing that the digital postprocessing, e.g. anti-alias filtering) is performed at a higher bit resolution.