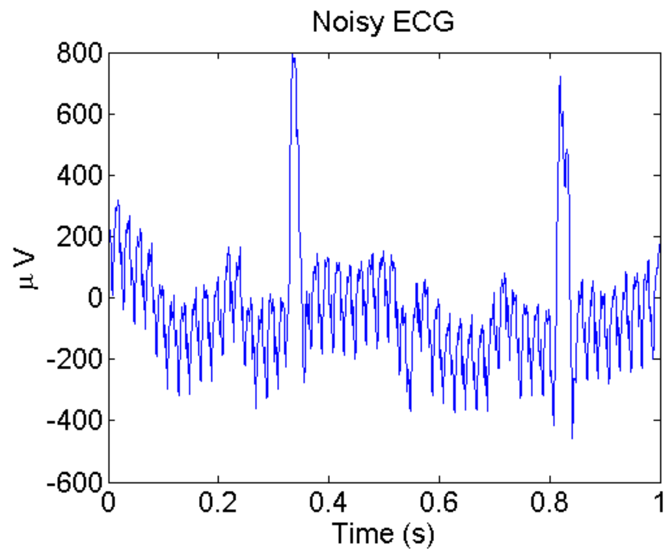
ENEL420 Advanced Signals

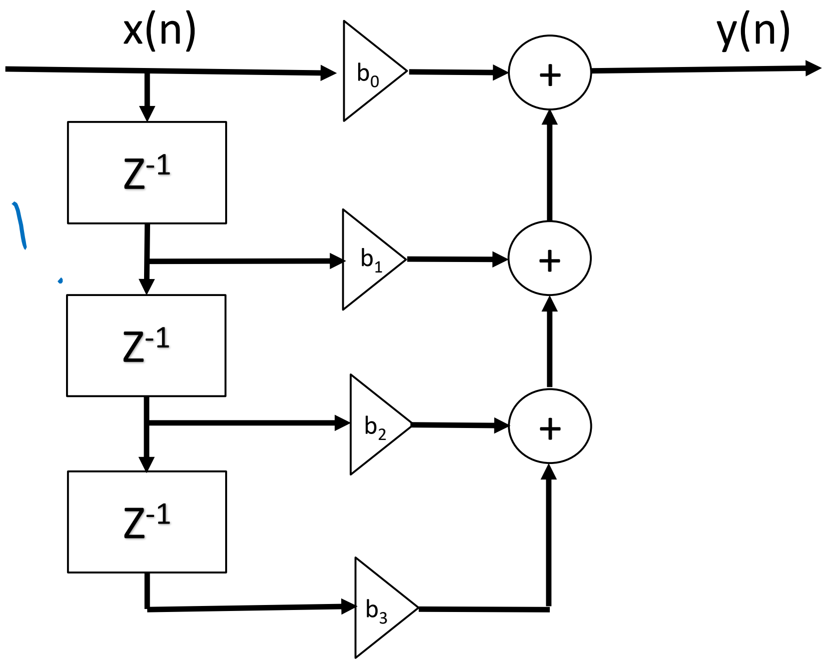
Introduction & Digital Filters

Richard Clare

2021



Nau Mai!



Lecturers

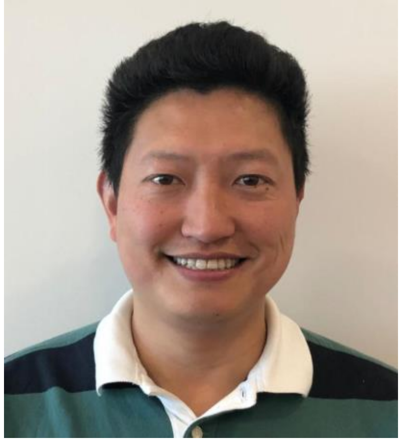
- Dr. Le Yang

  Course Co-ordinator

  le.yang@canterbury.ac.nz

  Link Building Room 510

  Teaching in Term 4



- Dr. Richard Clare

  richard.clare@canterbury.ac.nz

  Link Building Room 511

  Teaching in Term 3

Timetable*

- 3 lectures per week:
  Monday 2pm in E14
  Thursday 11am in E14
  Wednesday 10am in E14
- No tutorials and no laboratories for this course.

- * Always consult CIS for up-to-date schedule

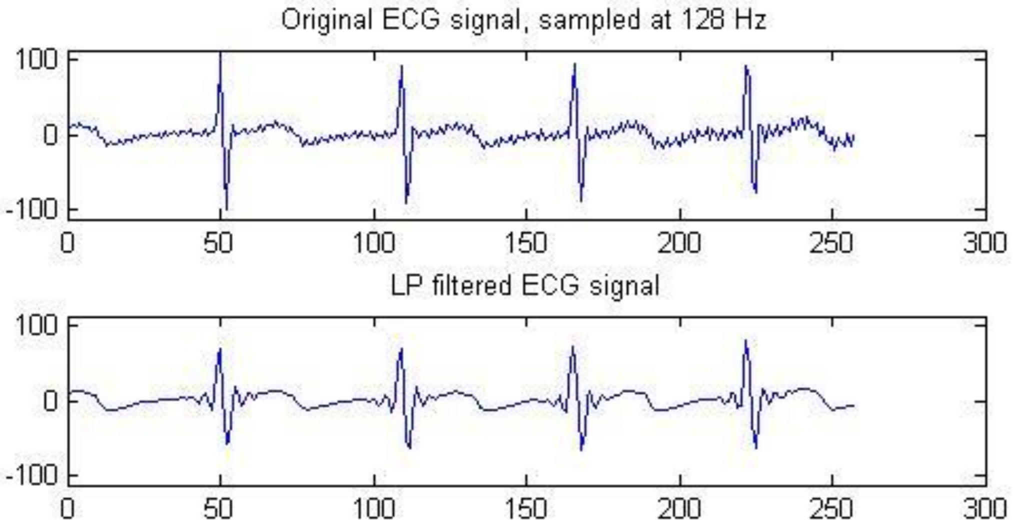
Assessments

- **Assignment 1** (2-person groups): Practical assignment using MATLAB or Python to filter signals. **Report due 5pm Friday 20 August**. Worth **15%.** (Dr. Clare)
- **Assignment 2** (2-person groups): Report on and demonstration of a specified signal processing topic, requiring some original investigation. **Progress report due 5pm Friday 17 September. Demonstrations during the week commencing 11 October. Final report due 5pm Friday 22 October**. Worth 30% (progress report 5%, demonstration 10% and final report 15%). (Dr Yang)
- **Final Examination (3 hr)** worth **55%**. The final exam covers the material from the whole semester.

Term 3 – Richard Clare 1

1. Sampled and discrete signals, filter design

Re-introduction to DSP. Sampling, aliasing, bandpass sampling and oversampling. FIR: symmetry and linear phase. FIR filter design: window, optimal and frequency sampling design methods. Finite wordlength effects.



Term 3 – Richard Clare 2

2. Multidimensional Fourier transform and applications

- Review of the 1D FT and DFT. Matrix and vector space representations of signals and transforms. The 2D FT and DFT. Applications: Reconstruction from projections, and image blurring and deconvolution.

Blurred image

WF, snr = 24 dB

Term 3 – Le Yang 1

- **3. Pattern Recognition**
- Review of basic probability distributions, linear algebra and convex optimization. Dimensionality reduction (singular value decomposition (SVD), principal component analysis (PCA), Fisher linear discriminant). Support vector machine (SVM), Gaussian mixture model (GMM) and Expectation-Maximization (EM).

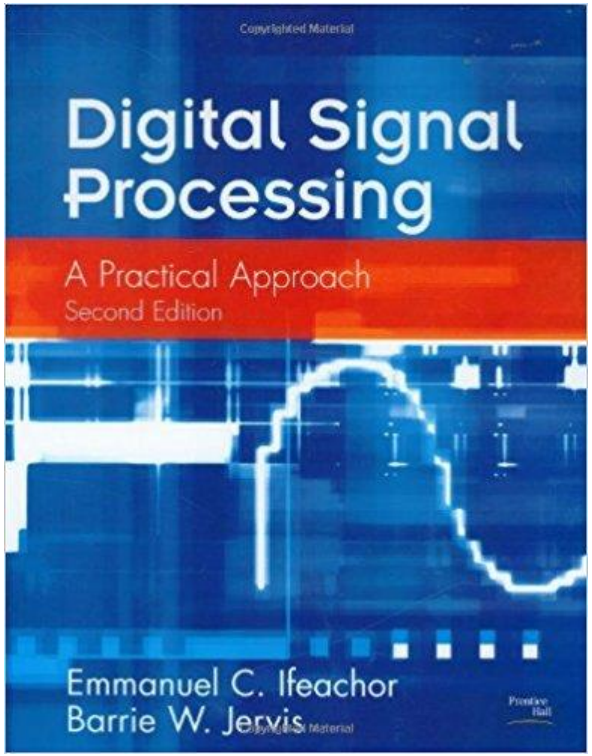
Term 4 – Le Yang 2

4. Statistical signal processing

- Detection theory. Neyman-Pearson (NP) theorem. Detection of random signals. Introduction to sparsity and compressive sensing (CS).

Textbook

- Digital Signal Processing, Ifeachor & Jervis, 2nd edition (2002).
- “Required Text”
- Covers 1/2 of the course. Three 2nd editions and three 1st editions in EPS library.
- Other references are given in the Course Outline (Learn).



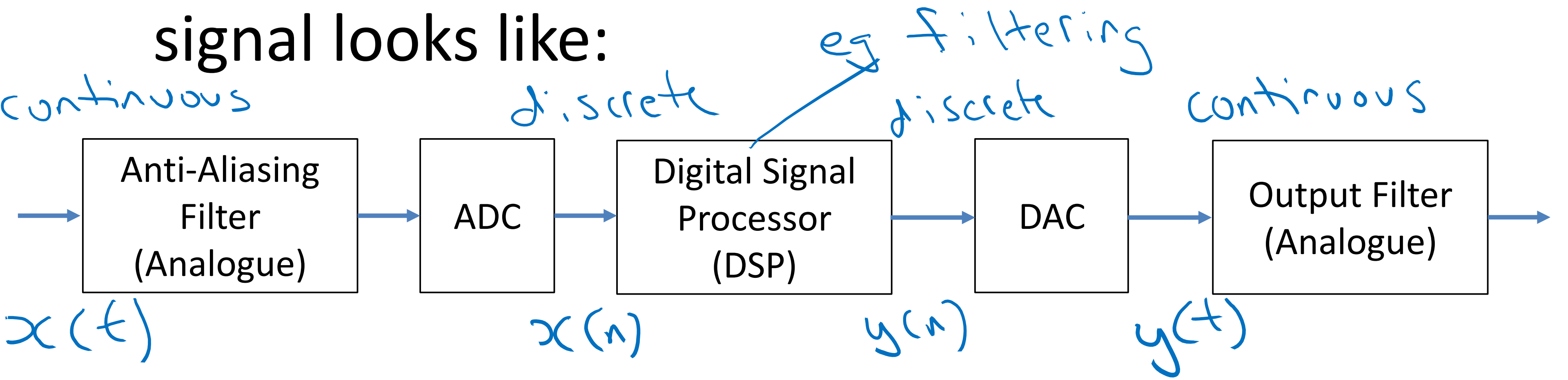
- I will put my lecture notes on Learn before the lectures, and the annotated notes after.

Tutorial Questions

- I will put up on Learn sets of tutorial problems for my part of course.
- The problems will be exam style questions.
- There is no timetabled tutorial session for this course.
- I will put up worked solutions at the end of my teaching block.

Digital Signal Processing (DSP)

- A typical DSP processing chain of an analogue signal looks like:



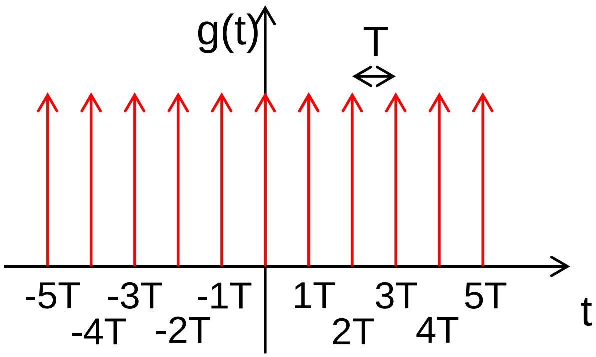
- ADC : Analogue to Digital Converter
- DAC : Digital to Analogue Converter
- We will now go through each of these blocks.

$f_s = \frac{1}{T}$

The Dirac Comb (Impulse Train)

- We can describe ideal sampling of a function g(t) by the ADC by considering a periodic series of Delta functions (Dirac comb):

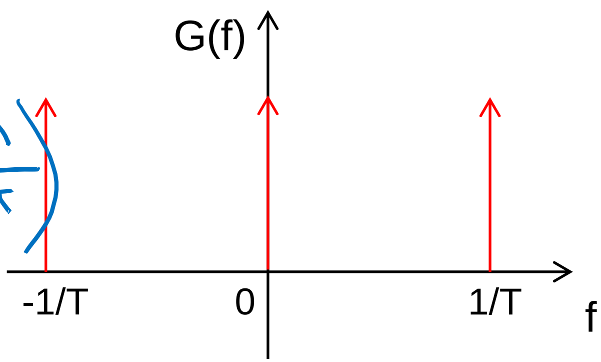
$$\Delta_T(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$$



- The Fourier transform of the Dirac comb is also a Dirac comb

$$\mathcal{F}\{\Delta_T(t)\} = \frac{1}{T}\sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T}\right)$$

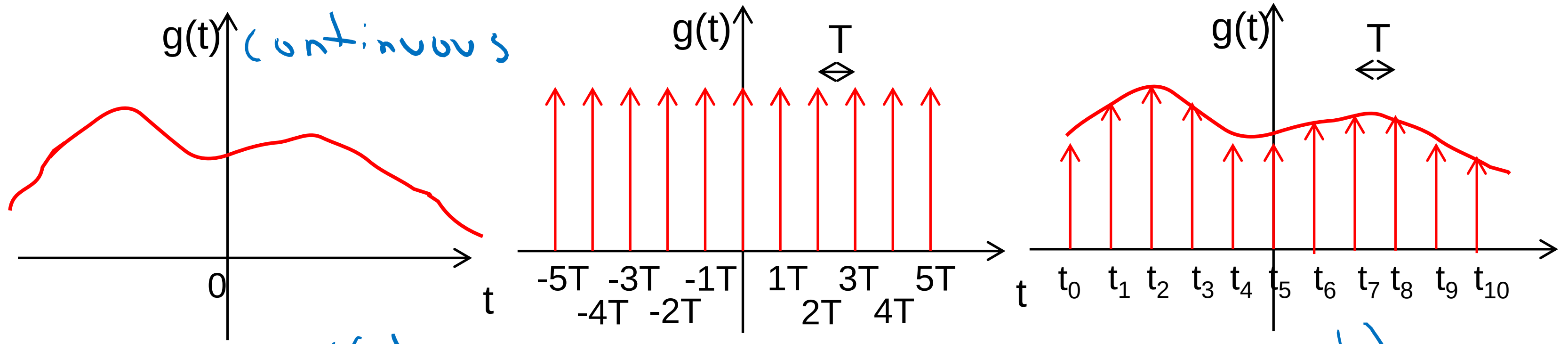
$$= \frac{1}{T}\Delta_{1/T}(f)$$



* convolution

Sampling

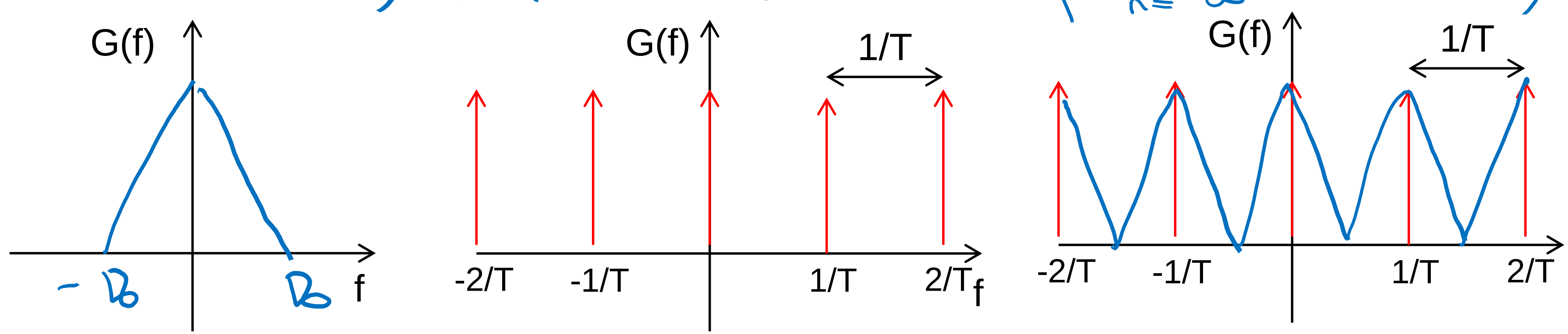
- Sampling in time domain is multiplication of the signal with a Dirac comb:



$$g(t) \times \Delta_T(t) = g(k) \text{ discrete}$$

- In frequency domain, it is a convolution:

$$G(f) * \Delta_{1/T}(f) = \frac{1}{T}\sum_{n=-\infty}^{\infty} G\left(f - \frac{n}{T}\right)$$



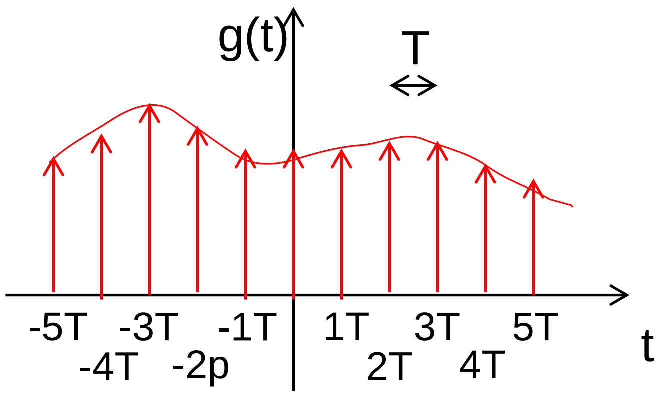
Nyquist-Shannon Sampling Theorem

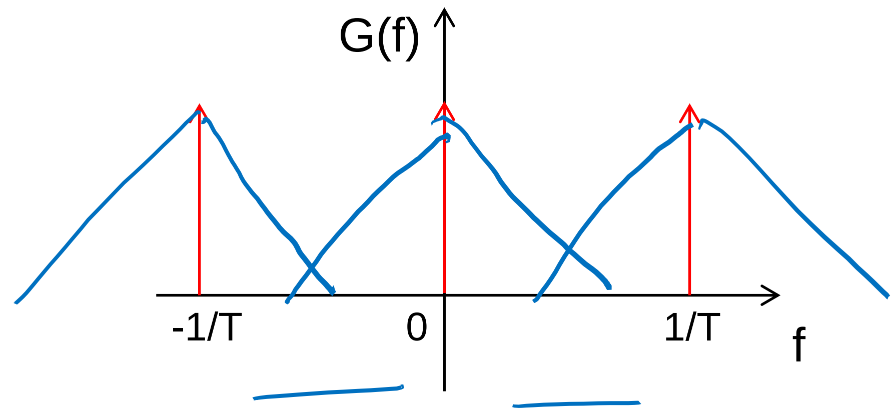
- The sampling rate must exceed twice the highest frequency in the signal g(t).

$$f_s = \frac{1}{T} \geq 2 f_{max}$$

maximum frequency component

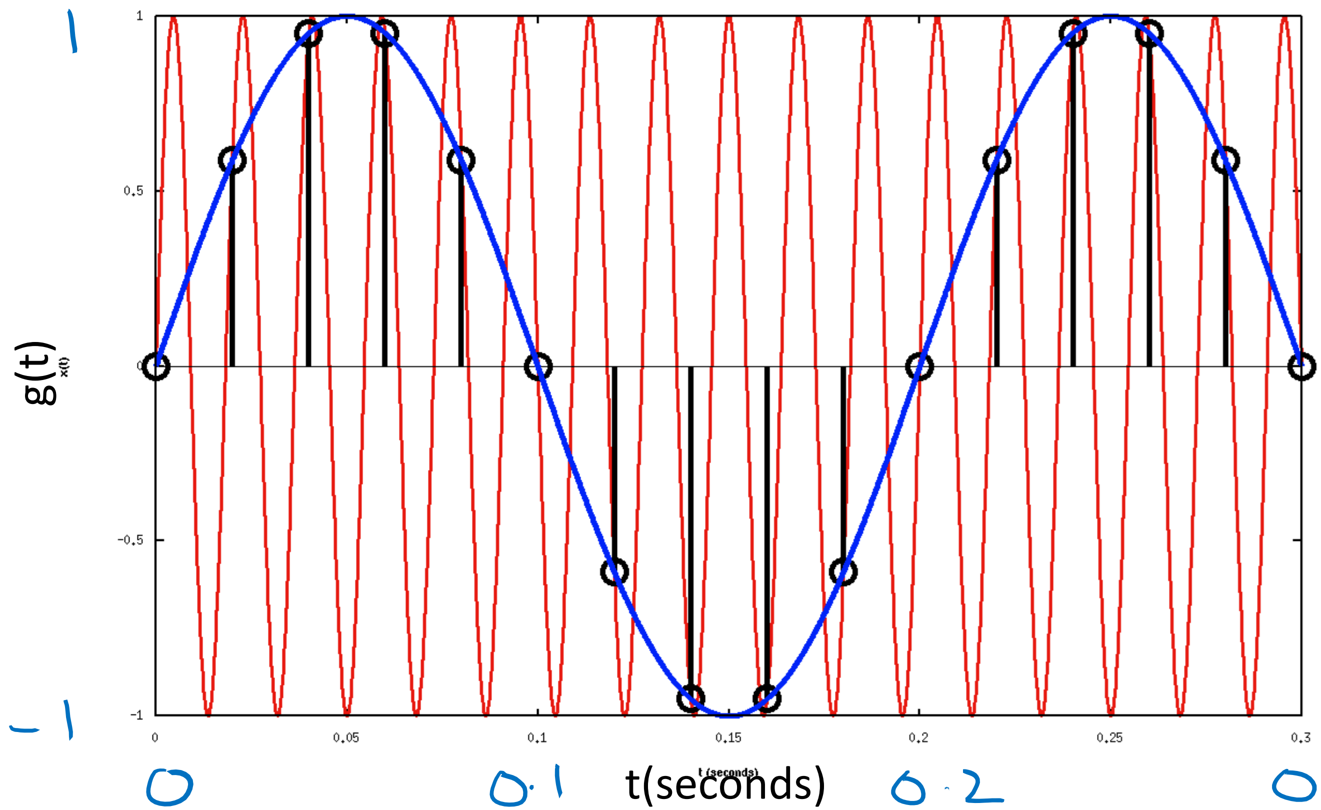
- Else if we sample below the Nyquist frequency, we will get aliasing (overlap in the Fourier domain):





Aliasing Example

o sample points



$f_{max} = 55$ Hz

$\Rightarrow$ f_s should be 110 Hz

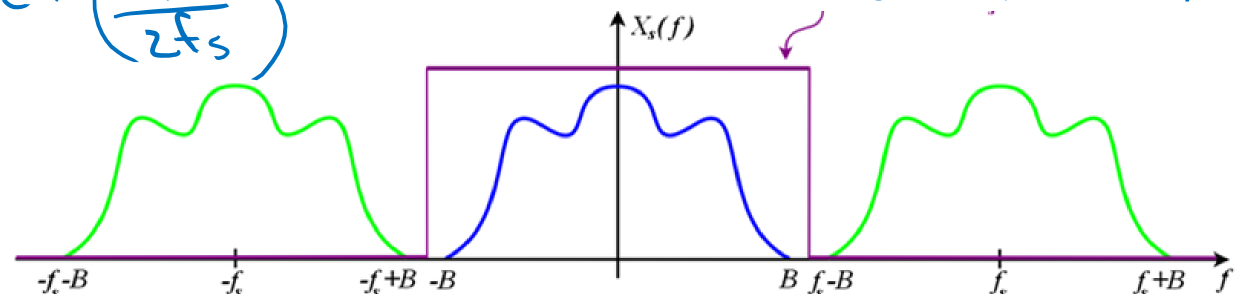
- Red=55 Hz true signal. T=0.02 s or f_s=50Hz
- Blue= 5Hz is an aliased signal.

Reconstruction of a Sampled Signal

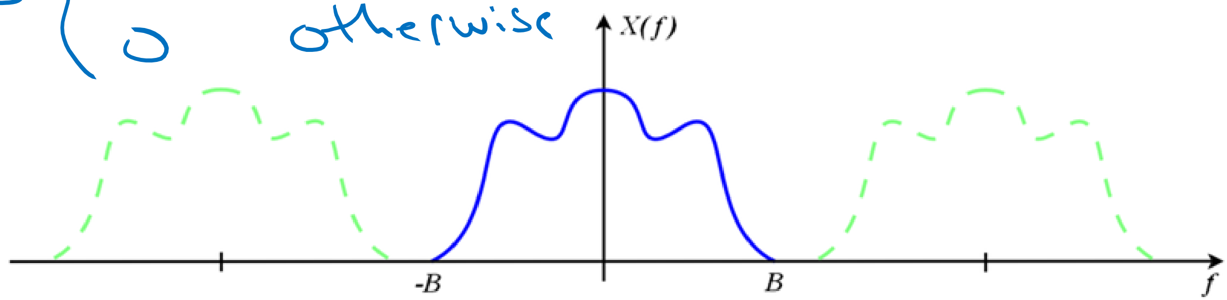
- Reconstruction is the determination of a continuous signal from a sequence of equally spaced samples.
- In theory, we can reconstruct a sampled signal by using a low pass filter to remove the unwanted spectra.

$$H(f) = \text{rect}\left(\frac{f}{2f_s}\right)$$

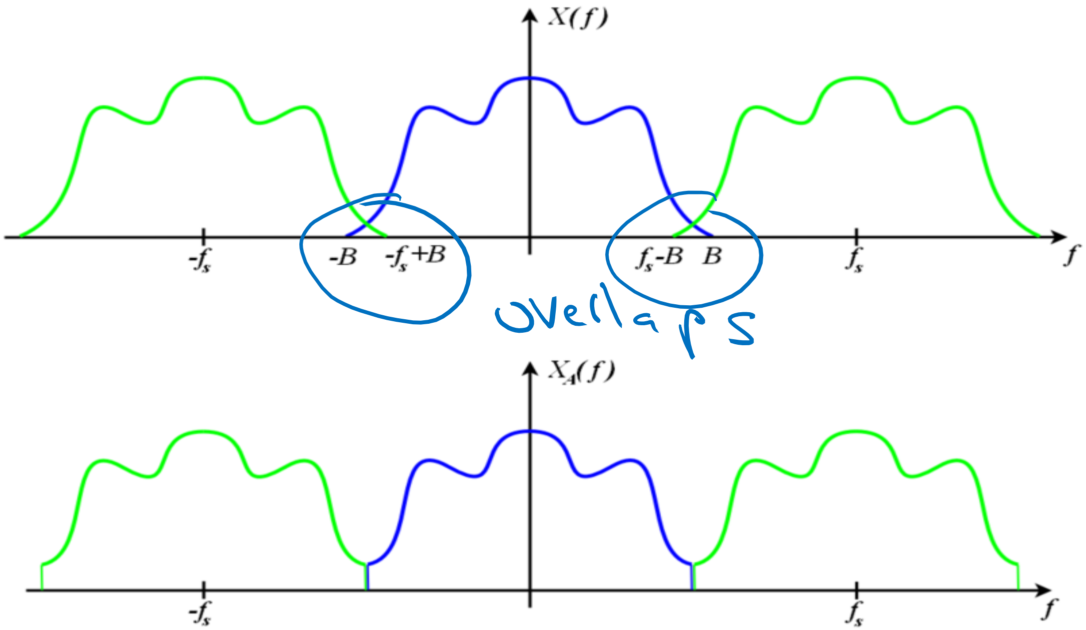
Low Pass filter



$$\text{rect}(x) = \begin{cases} 1 & |x| \le 1/2 \\ 0 & \text{otherwise} \end{cases}$$



Reconstruction if not Nyquist Sampled



- Two problems:

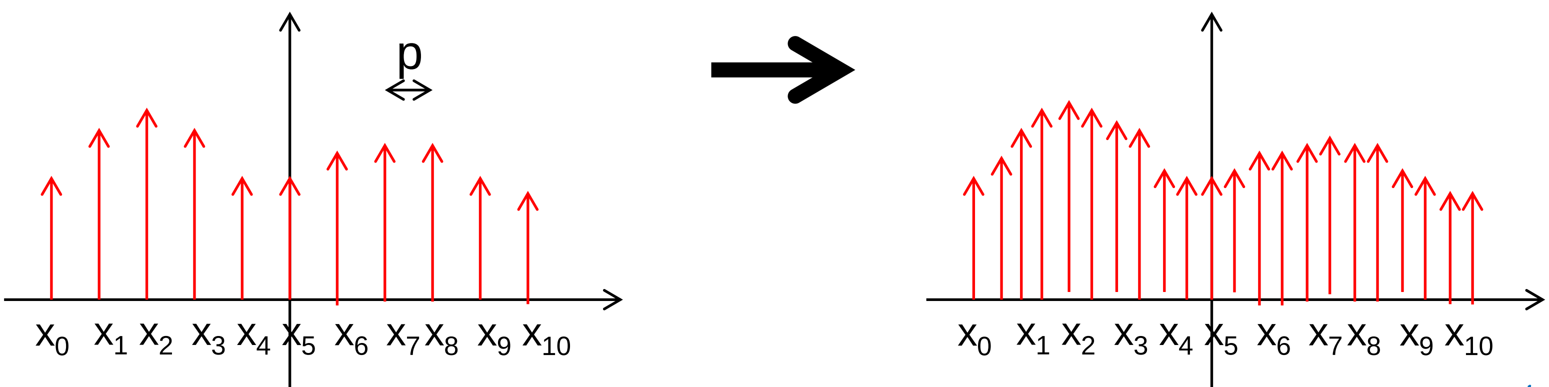
1. There are high frequencies that we cannot measure.
2. Frequencies below the Nyquist limit are distorted.

- It is no longer possible to isolate the original signal just by filtering.

Interpolation

- Reconstruction in the time domain can be considered as interpolation.
- Interpolation can be written as a convolution with a kernel function:

$$f(k_0) = \sum_{k} \underbrace{f(k)}_{\text{data}} \underbrace{K(k_0 - k)}_{\text{interpolation kernel}}$$

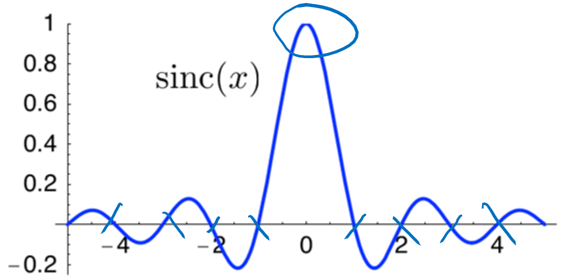


f(k) data

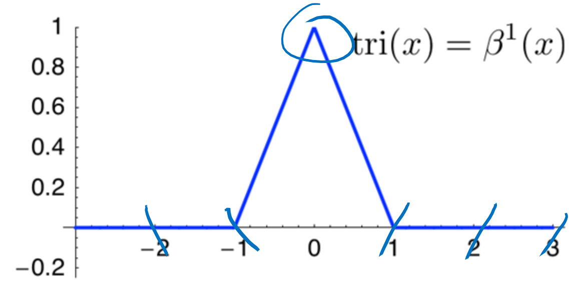
$f(k_0)$ interpolated signal

Interpolation Kernels

- Bandlimited



- Piecewise linear

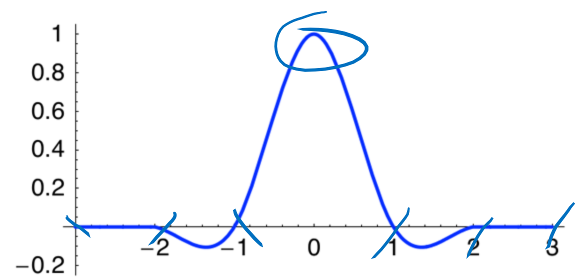


Interpolation condition

$$K = \begin{cases} 1 & k=0 \\ 0 & \text{other values of } k \end{cases}$$

To return the original values.

- Cubic convolution

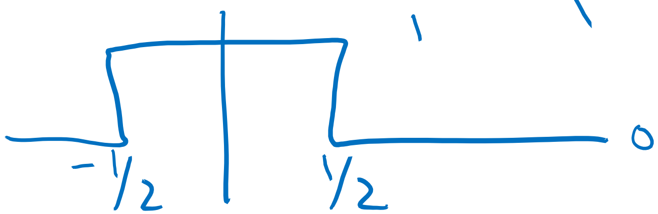


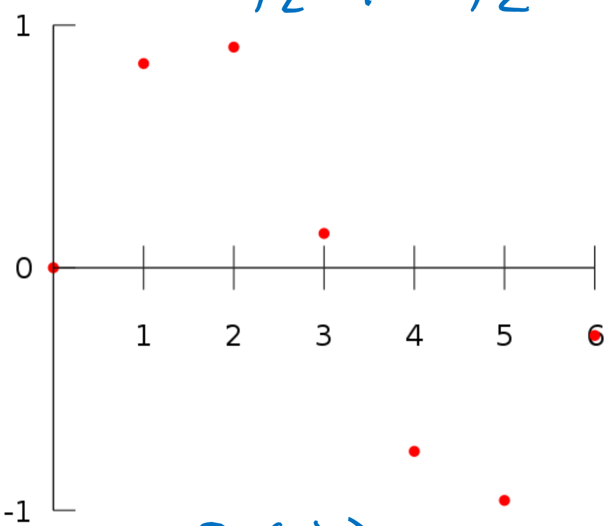
[Keys, 1981; Karup-King 1899]

Nearest Neighbour Interpolation

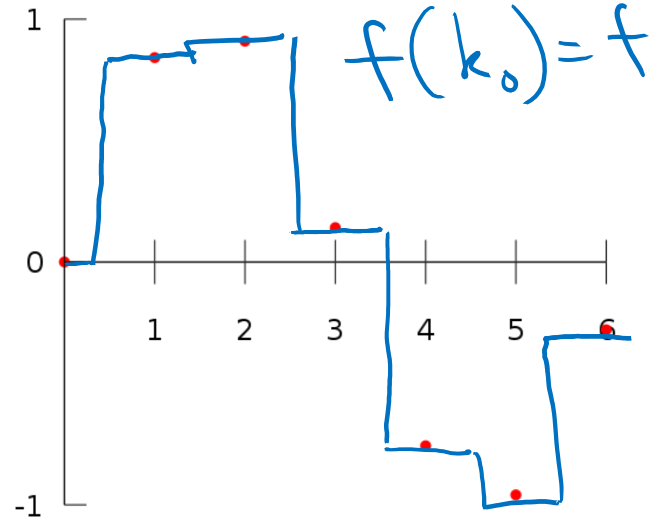
- Kernel is the rect function

$$K(x) = \mathrm{rect}(x) = \begin{cases} 1 & |x| \le 1/2 \\ 0 & \text{otherwise} \end{cases}$$





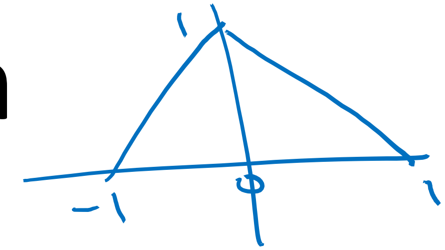
$f(k)$



$$f(k_0) = f(k) * K$$

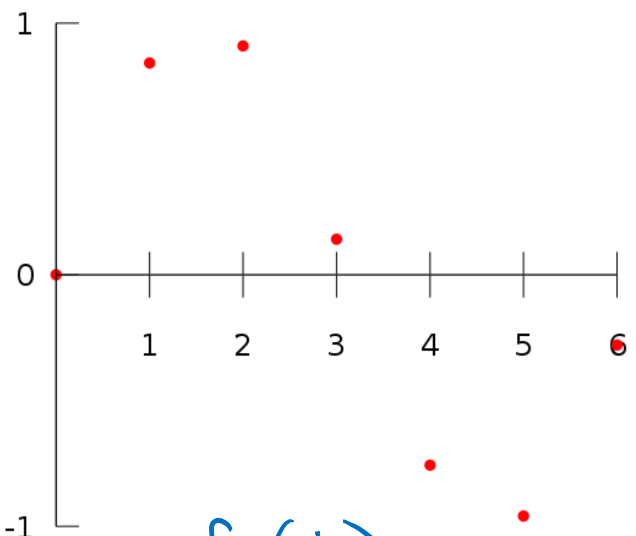
$f(k_0)$

Linear Interpolation

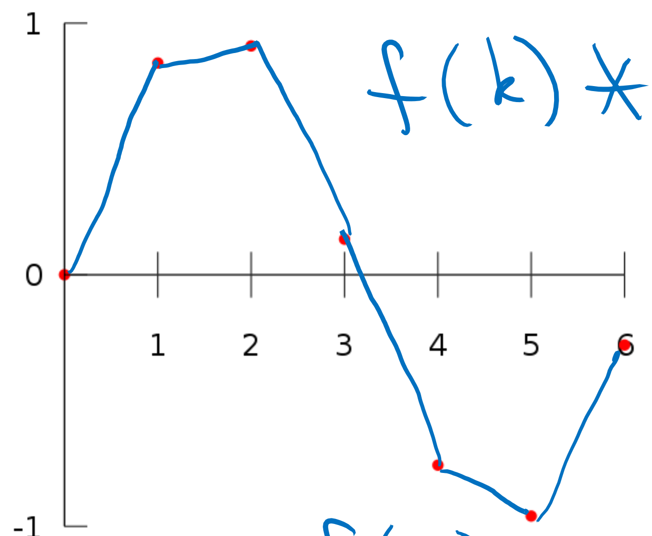


- Kernel is the tri function

$$K(x) = \mathrm{tri}(x) = \begin{cases} 1 - |x| & |x| < 1 \\ 0 & \text{otherwise} \end{cases}$$



$f(k)$



$f(k) * K$

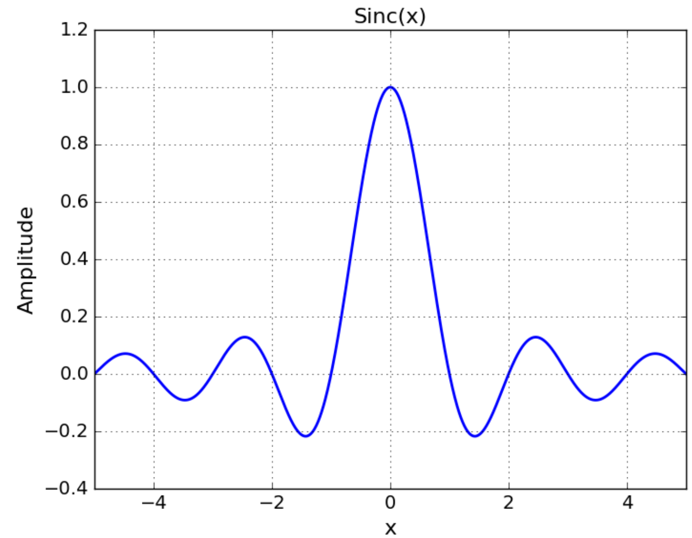
$f(k_0)$

Sinc Interpolation

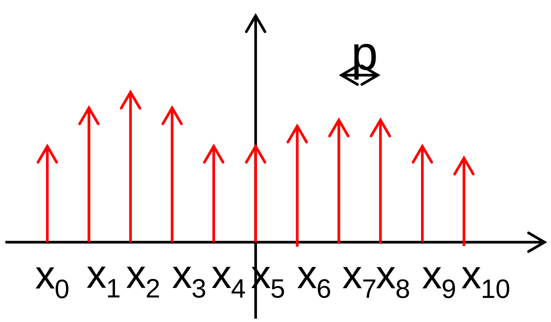
- Kernel is the sinc function
- Sinc interpolation can perfectly reconstruct an analogue signal from its sampled signal if:

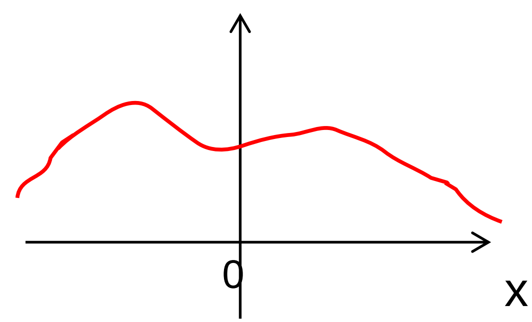
Equivalent

1. it was sampled at the Nyquist frequency or higher
2. the signal is band-limited up to the Nyquist frequency
3. there is no aliasing.



$$K(x) = \mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$$





Zero Padding

interpolated signal

sampled data

sinc kernel

- Let's consider the sinc interpolation case:

$$f(x) = \sum_k f(k)\,\mathrm{sinc}(x-k)$$

$\downarrow$ F.T.

$$F(u) = \mathcal{F}\left\{\sum_k f(k)\,\mathrm{sinc}(x-k)\right\}$$

$\downarrow$ linearity

$$= \sum_k f(k) \cdot \mathcal{F}\left\{\mathrm{sinc}(x-k)\right\}$$

$\downarrow$ shifting

$$F(u) = \boxed{\sum_k f(k)\, e^{-j2\pi k u}} \cdot \mathrm{rect}(u)$$

Discrete Fourier Transform

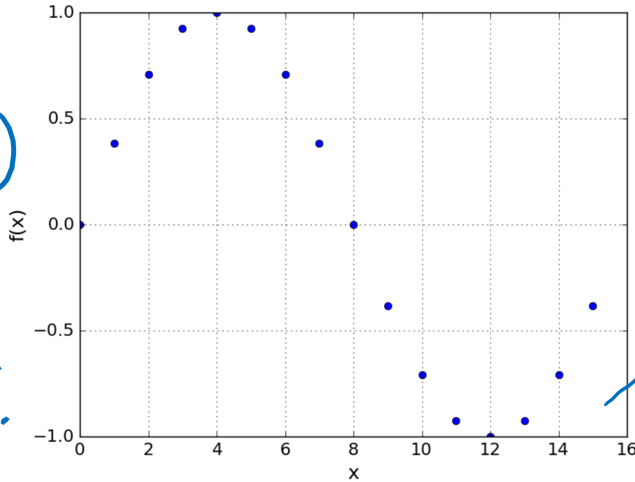
D.F.T

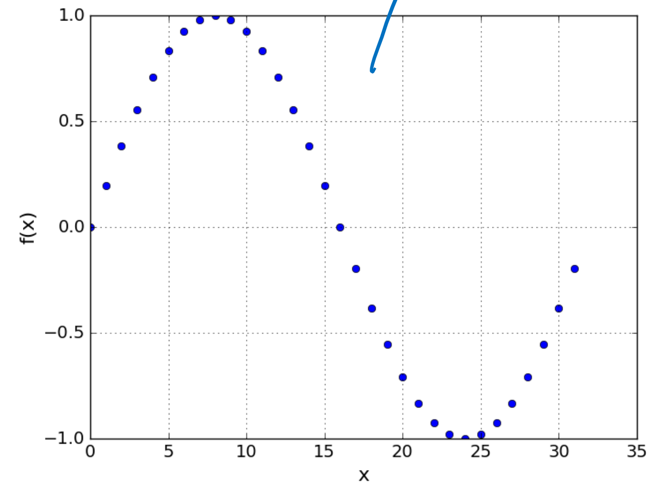
- Zeropadding of the Fourier Transform of f(k) is equivalent to sinc interpolation of f(k)

Zero Padding Example

Interpolated Signal

f(x)



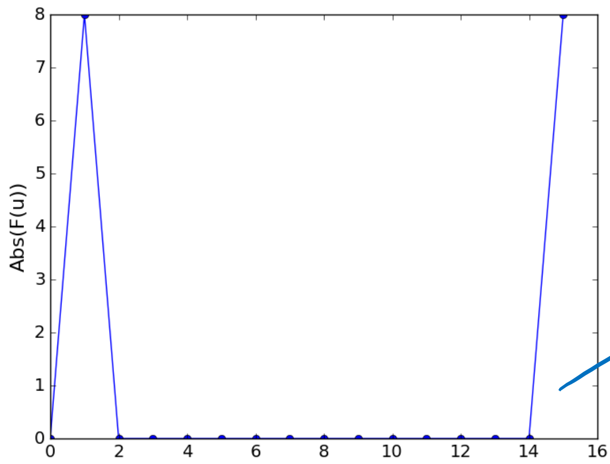


F.T.

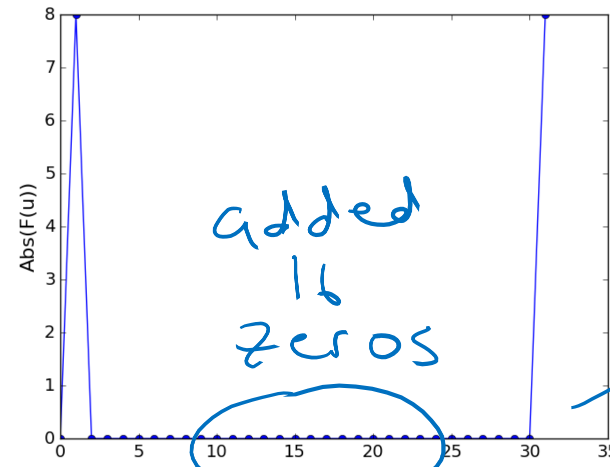
15 data points

FFT

F(u)



Zeropad



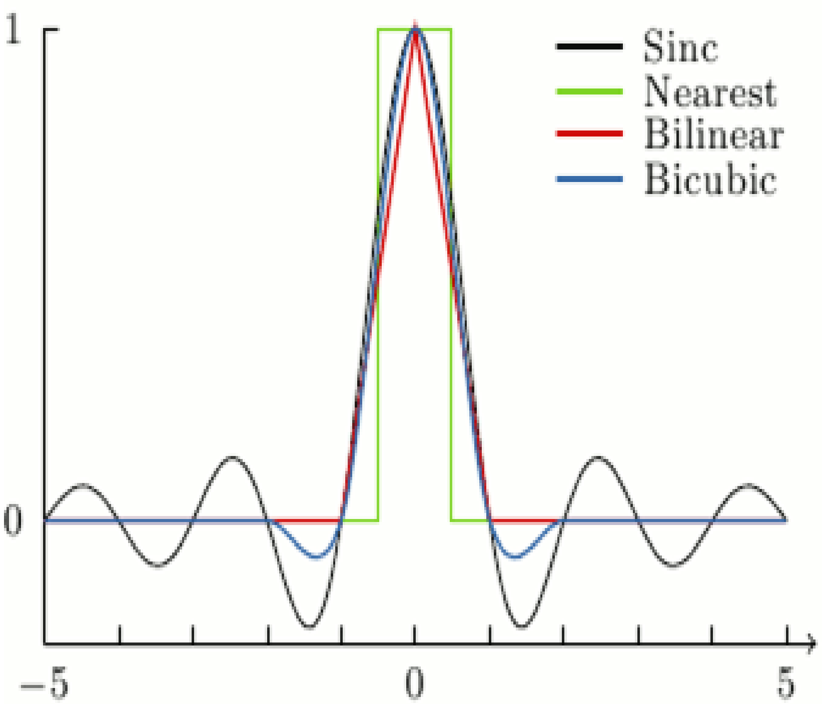
15 data points

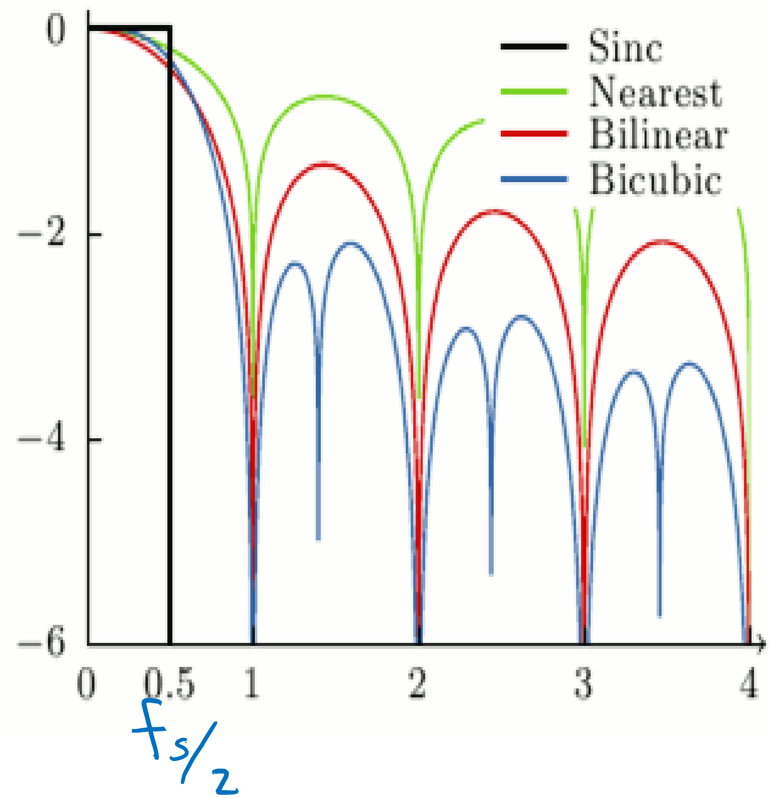
added 16 zeros

32 data points

Kernels $K(x) \xrightarrow{F.T} |\mathcal{F}\{K(x)\}|$

Frequency Behaviour



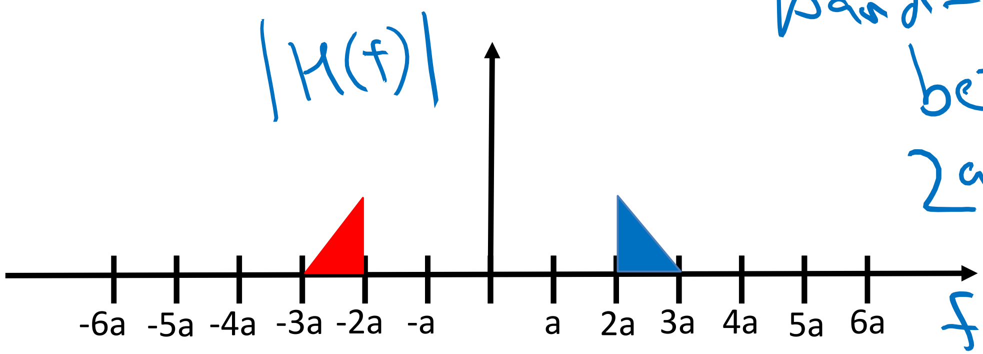


- Compared to sinc, other interpolators allow frequencies higher than $f_s/2$ into the analogue output, and attenuates frequencies less than $f_s/2$ unevenly.

Sampling Bandpass Signals

- The sampling theorem (Nyquist) is stated for lowpass signals, but is uneconomical for bandpass signals. Consider the following bandpass signal:

band-limited between 2a - 3a



- According to Nyquist, the signal should be sampled at 6a, which is much faster than it needs to be.

$f_{max} = 3a \quad \therefore \quad f_s = 2 \times f_{max} = 6a$

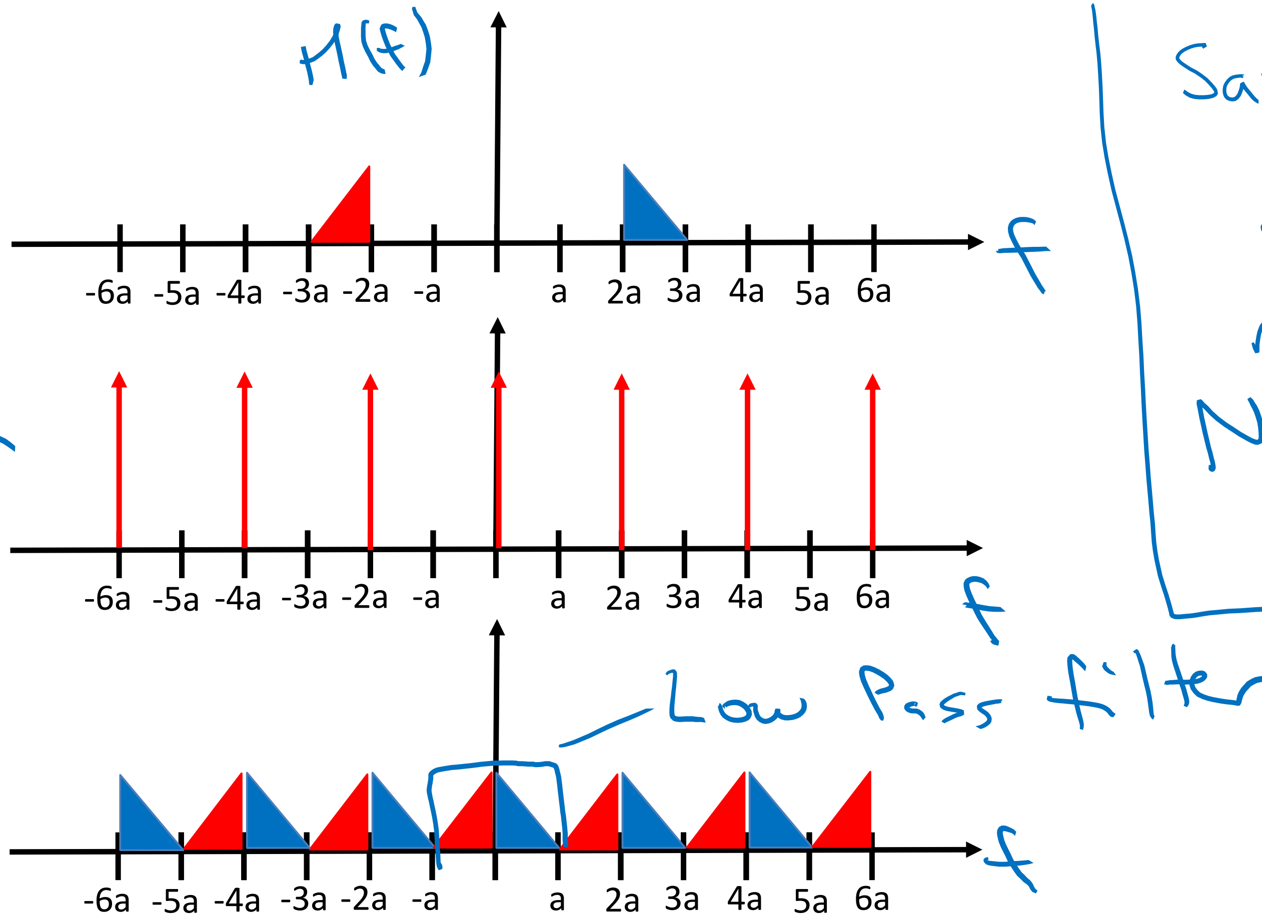
sifting property $h(t) * \delta(t-t_0) = h(t-t_0)$

Bandpass Sampling Approach

- If we sample at a carefully chosen sampling frequency much lower than the Nyquist rate, we can still avoid aliasing. Consider:

Convolution $*$

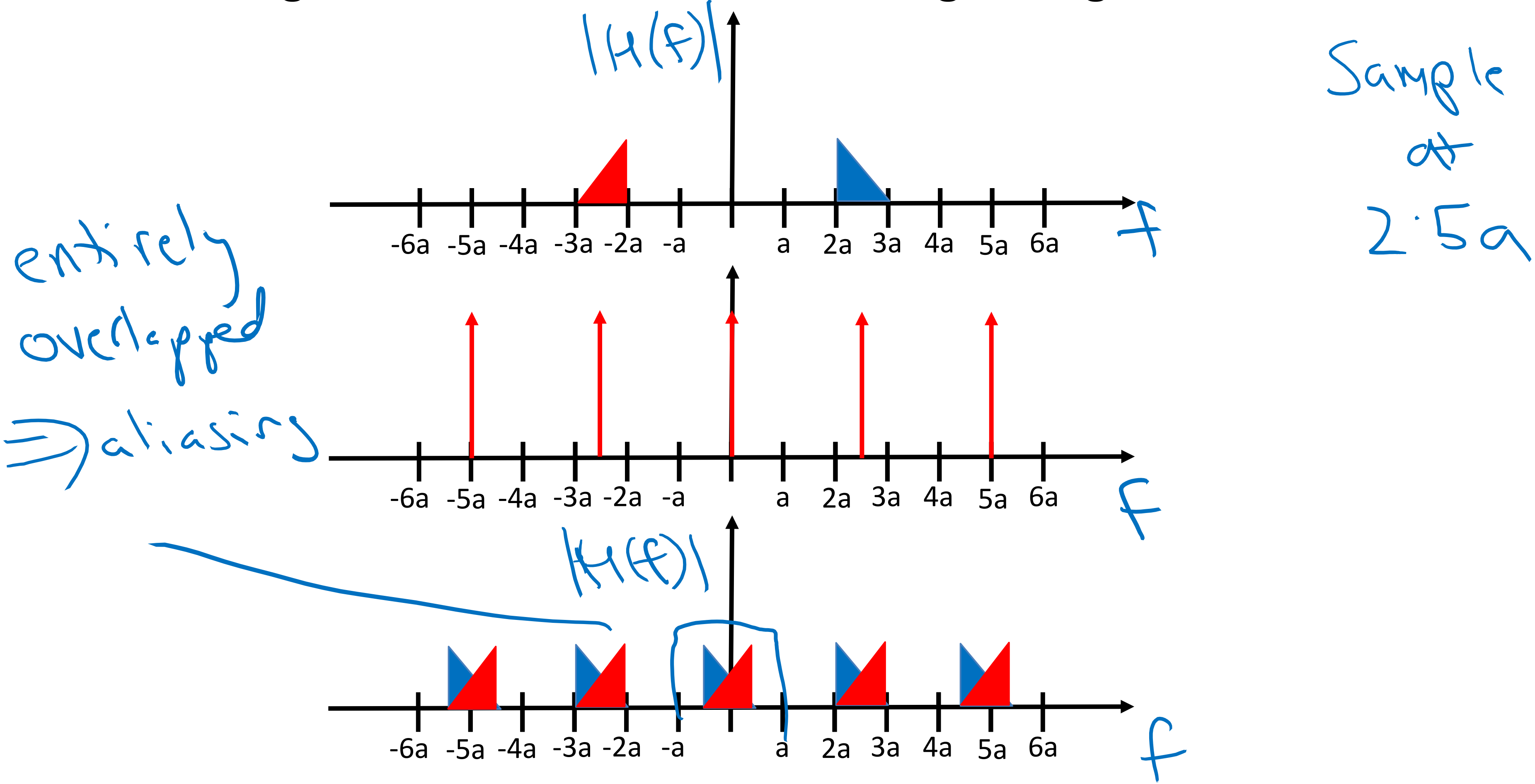
No aliasing



Sample at 2a not Nyquist 6a

Wrong Sampling Frequency

- If we choose the wrong sampling frequency, we will get aliasing, and we can't recover the original signal.



Sampling Frequencies

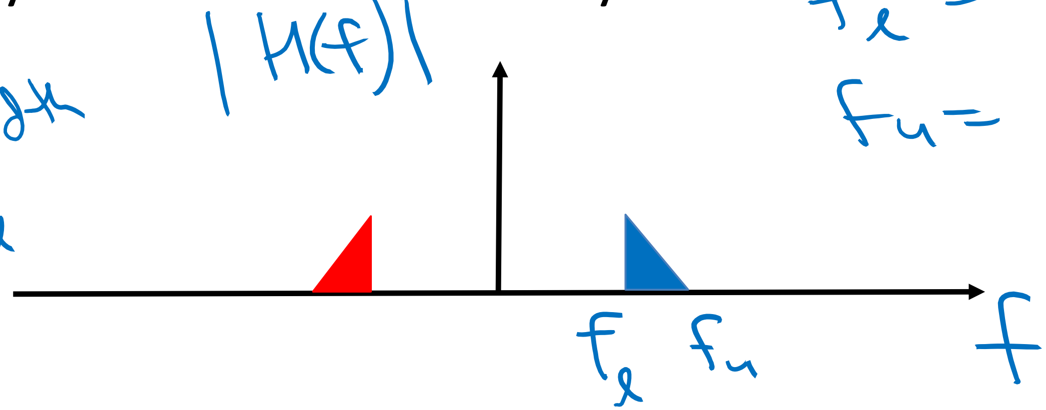
- Aliasing will occur with bandpass sampling if the sampling frequency is not chosen carefully.

f_ℓ = lower frequency

f_u = upper frequency

B = bandwidth

$B = f_u - f_\ell$



- For the the general case, in order to avoid aliasing

$$\frac{2f_u}{n} < f_s < \frac{2f_\ell}{n-1}$$

where

$$1 \le n \le I_{MLT}\left(\frac{f_u}{B}\right)$$

I_{MLT} = maximum integer less than

Sampling frequencies 2

- The acceptable rage of sampling frequencies are shown in white, and unacceptable (aliasing) in grey.

No aliasing example

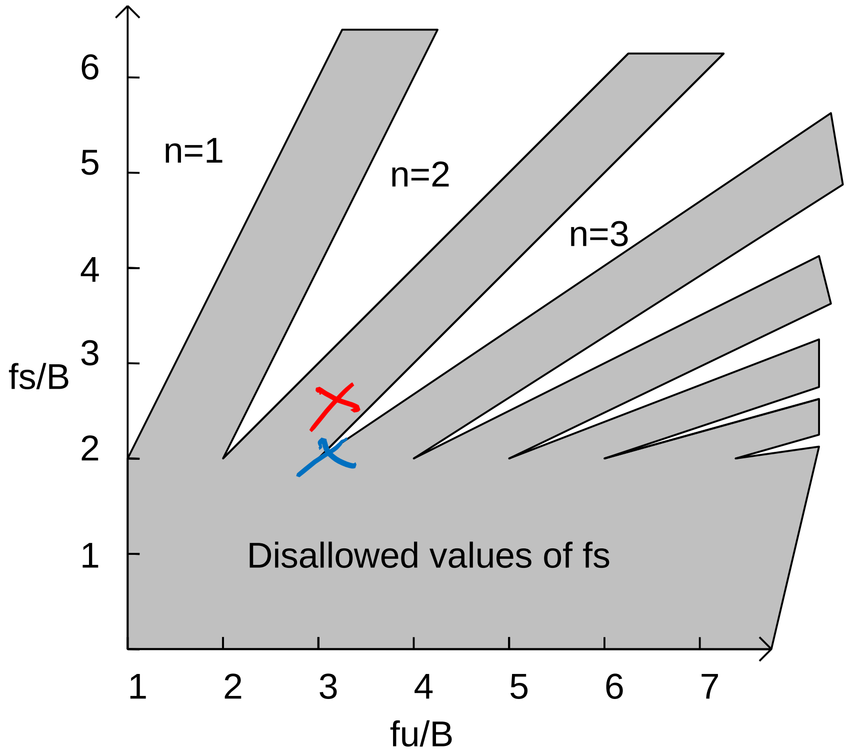
$f_u = 3a$

$B = a$

$\frac{f_u}{B} = 3$

$f_s = 2a$

$\frac{f_s}{B} = 2$



Aliasing example

$\frac{f_u}{B} = 3$

$f_s = 2.5a$

$\frac{f_s}{B} = 2.5$

Sampled signal

$T = \frac{1}{f_s}$

Z-transform

Dirac comb

- Our signal sampled with a period T is given by:

$$f_s(t) = \sum_{n=0}^{\infty} f(nT)\,\delta(t-nT)$$

- If we take the Laplace Transform of both sides:

$$\mathcal{L}\{f_s(t)\} = F(s) = \sum_{n=0}^{\infty} f(nT)\,e^{-nTs} \quad \Big|\, \text{delay}$$

$$= \sum_{n=0}^{\infty} f(nT)\,(e^{sT})^{-n}$$

- If we make the substitution $z = e^{sT}$ we arrive at the one-sided (causal) Z-transform

$f(k) = f(nT)$

$$F(z) = \sum_{k=0}^{\infty} f(k)\,z^{-k}$$

samples

Green = stable

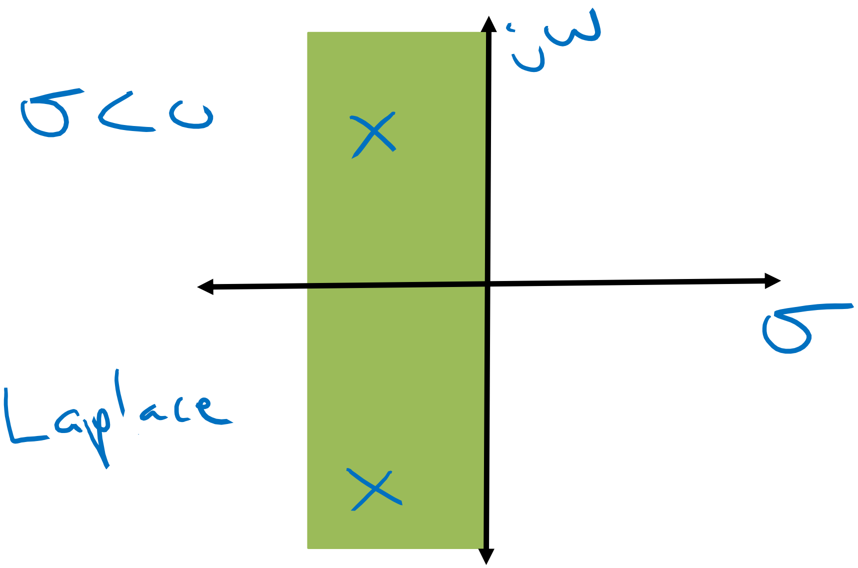
Z-transform Stability

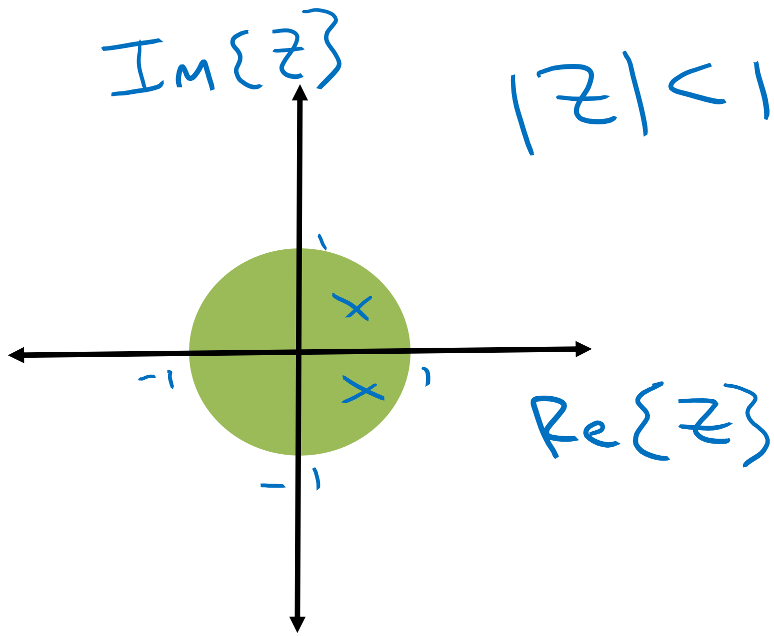
- For a continuous system to be stable, the poles must lie in the left hand s-plane (σ <0).

$Z = e^{\sigma T}$ $S = \sigma + j\omega$

$Z = e^{\sigma T} \cdot e^{j\omega T}$

- This corresponds to a Z-transform having all poles inside the unit circle.





Z-transform Stability 2

- Consider the following 2 transfer functions:

1) $H(z) = \dfrac{z}{z - 0.9}$

pole at $z = 0.9$

2) $H(z) = \dfrac{z}{z+2}$

pole at $z = -2$

$$\boxed{\mathcal{Z}\{\alpha^n u(n)\} = \frac{z}{z-\alpha}}$$

Impulse response $h(n)$

$h(n) = (1, -2, 4, -8, \ldots)$

stable

unstable

$h(n) = \{1, 0.9, 0.81, 0.729 \ldots\}$

Integrator & Differentiator

- A digital integrator adds a new value on to the sum of preceding values:

Difference equation

$$y(n) = x(n) + y(n-1)$$

Z transform

$$Y(z) = X(z) + Y(z)z^{-1}$$

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

- A digital differentiator outputs the difference between two successive samples:

difference equation

$$y(n) = x(n) - x(n-1)$$

Z transform

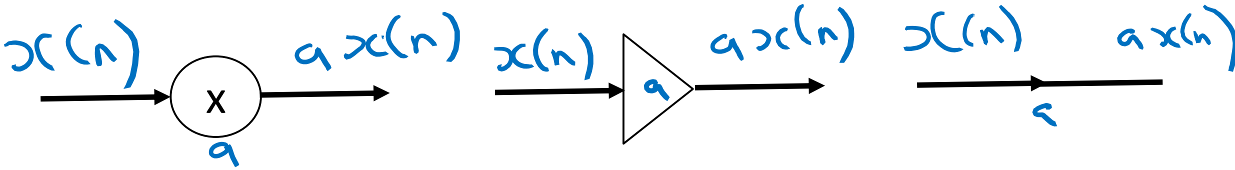
$$Y(z) = X(z) - X(z)z^{-1}$$

$$Y(z) = X(z)(1 - z^{-1})$$

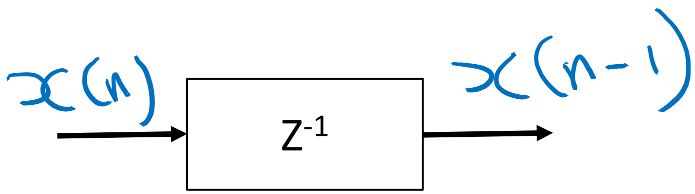
Digital Filter Elements

- We can use the Z-transform to easily implement digital filters:

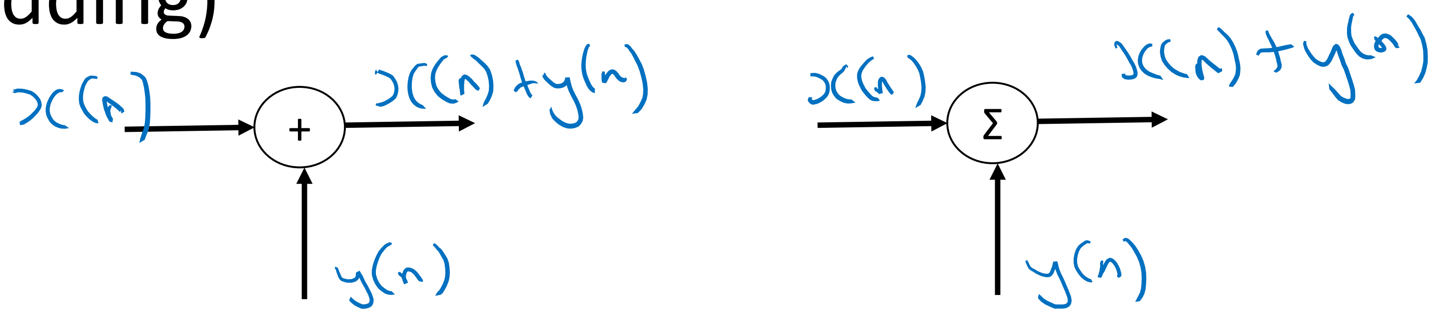
1) Scaling (gain)



2) Shift (delay)

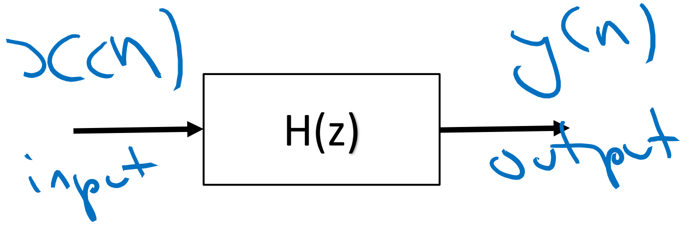


3) Sum (adding)



Digital Filter Transfer Functions

- We can use these three filter elements to realise any filter of the form:



$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{K} b_k z^{-k}}{1 + \sum_{m=1}^{M} a_m z^{-m}}$$

- By convention, the first term in the denominator is always 1 to avoid having multiple ways of expressing the same transfer function.

Min/Max error are $\pm \Delta/2$

Quantization Noise

- Quantization generates noise due to the error in not being able to represent the continuous signal perfectly.

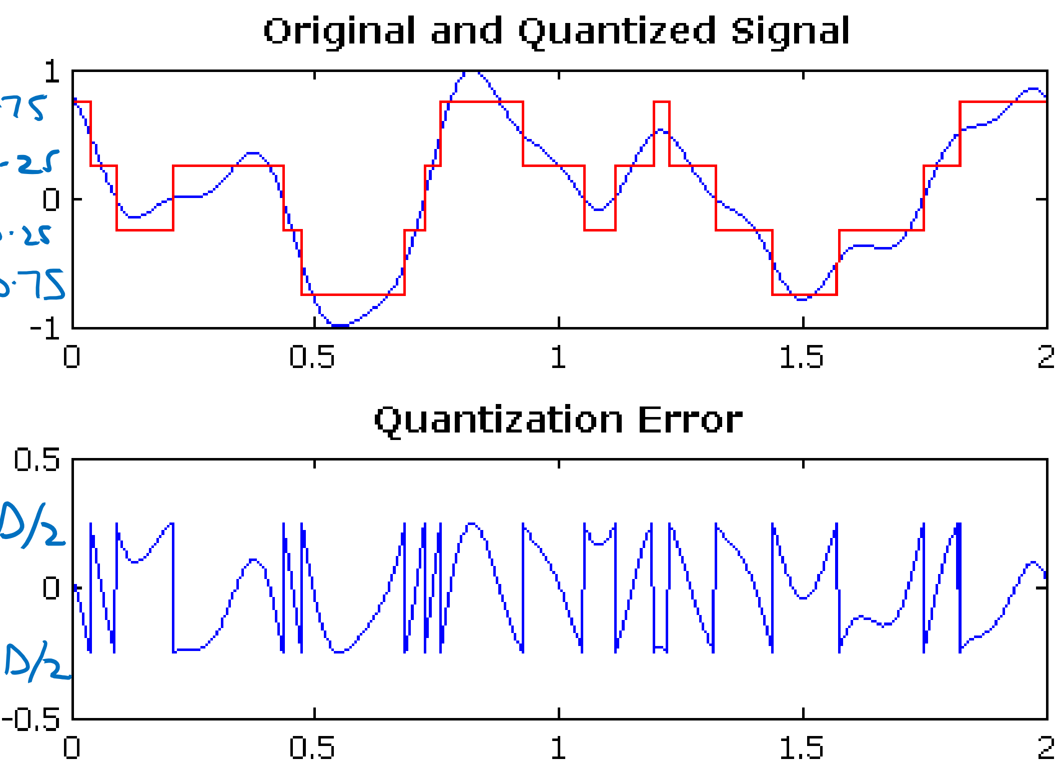
$4 = 2^2$ levels

2 bits

Blue: continuous signal

Red: quantized signal

0.75
0.25
-0.25
-0.75



$\Delta/2$

$-\Delta/2$

step size or resolution

$\Delta = 0.5$

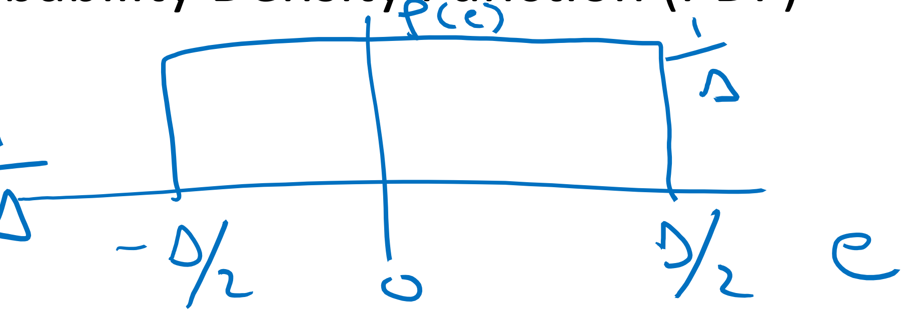
Error = original signal − quantized signal

RMS quantization noise $\sigma = \frac{\Delta}{\sqrt{12}} = \frac{\Delta}{2\sqrt{3}}$

Quantization Noise 2

- Assume the quantization error is uniformly distributed with maximum error=Δ/2. The Probability Density Function (PDF) of quantization noise is then:

$$\sum \text{all prob.} = 1 = \left(\frac{\Delta}{2} + \frac{\Delta}{2}\right) \cdot \frac{1}{\Delta}$$



Variance (= power) $\sigma^2 = \int (x-\mu)^2 f(x)\,dx$ $\mu = 0$ here

$$\sigma^2 = \int_{-\Delta/2}^{\Delta/2} e^2 \cdot f(e)\,de \qquad f(e) = \frac{1}{\Delta}$$

$$\sigma^2 = \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} e^2\,de = \frac{1}{\Delta}\left[\frac{e^3}{3}\right]_{-\Delta/2}^{\Delta/2}$$

$$\sigma^2 = \frac{1}{\Delta}\left[\frac{\Delta^3}{24} + \frac{\Delta^3}{24}\right] = \frac{\Delta^2}{12}$$

Quantization Noise 3

- Quantization is a trade-off between clipping, which occurs at the signal peaks, and quantization noise, which occurs throughout the signal.
- If we increase the number of bits by one:

$$\Delta \Rightarrow \Delta/2$$

noise power $\propto \Delta^2$

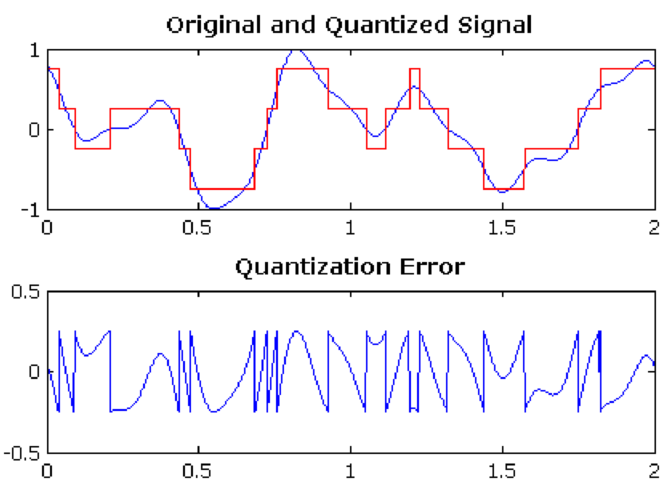
decreases by $\left(\frac{1}{2}\right)^2 = \frac{1}{4}$

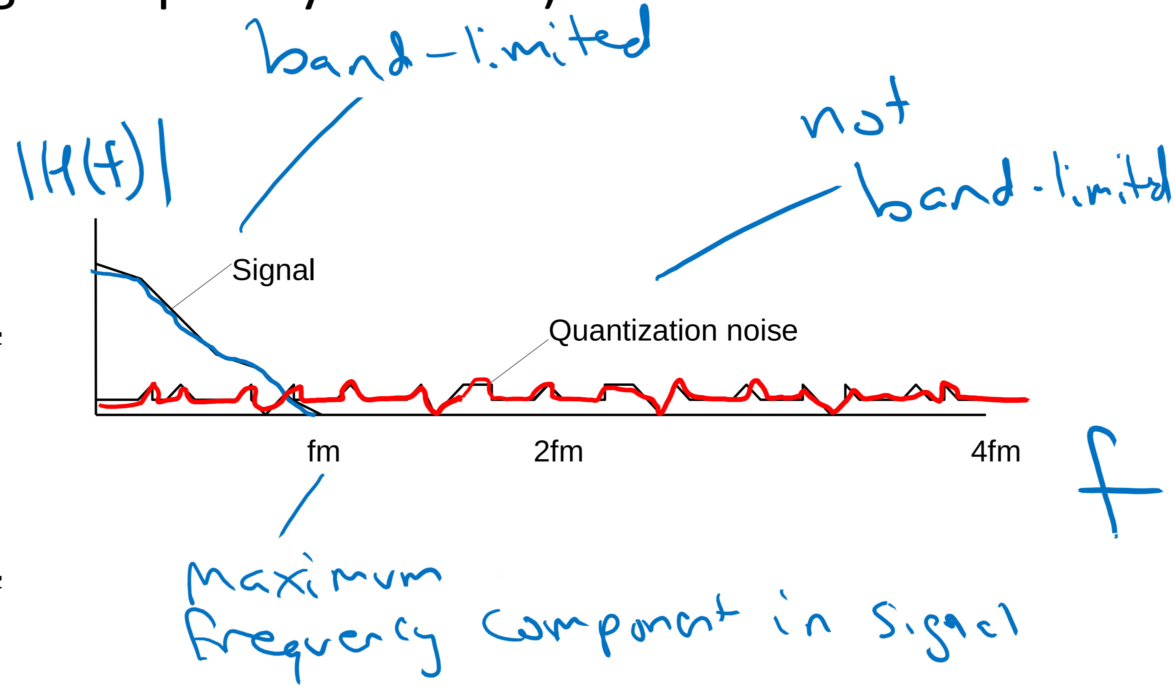
$$10\log_{10}(1/4) = 6\,\text{dB}$$

- Each additional bit of precision increases the signal-to-noise ratio (SNR) by 6 dB.

Oversampling

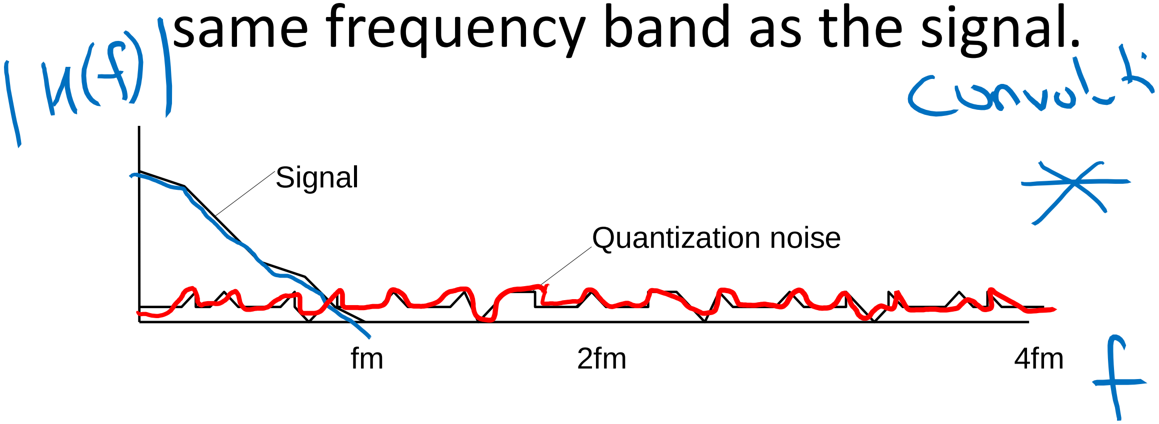
- Oversampling is an approach in DSP to reduce the effects of quantization noise.
- The original signal may be band-limited, but the noise introduced by quantization is not band-limited (the sharp transitions indicate high frequency content).





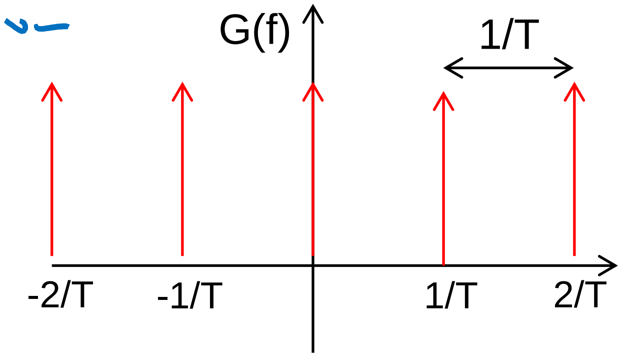
Oversampling 2

- Nyquist sampling at $2f_m$ aliases the quantization noise to the same frequency band as the signal.

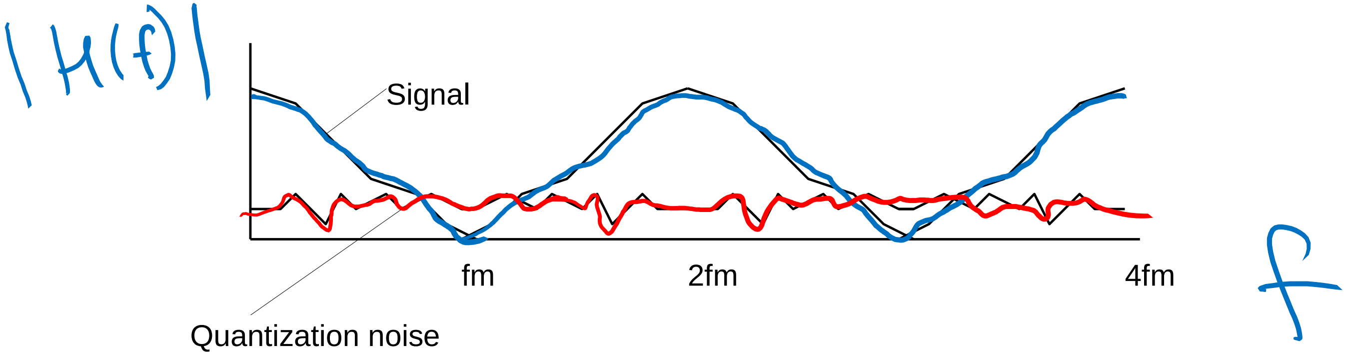


convolution

$*$



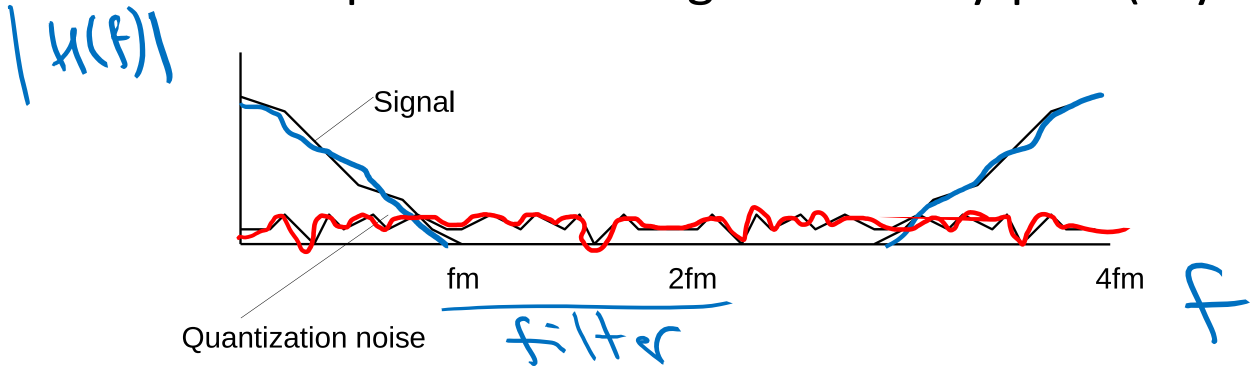
- We cannot remove the noise by filtering without also filtering the signal as they now share the same frequency range.



Oversampling 3

2xNyquist

- If instead we sample at a rate higher than Nyquist (say $4f_m$):



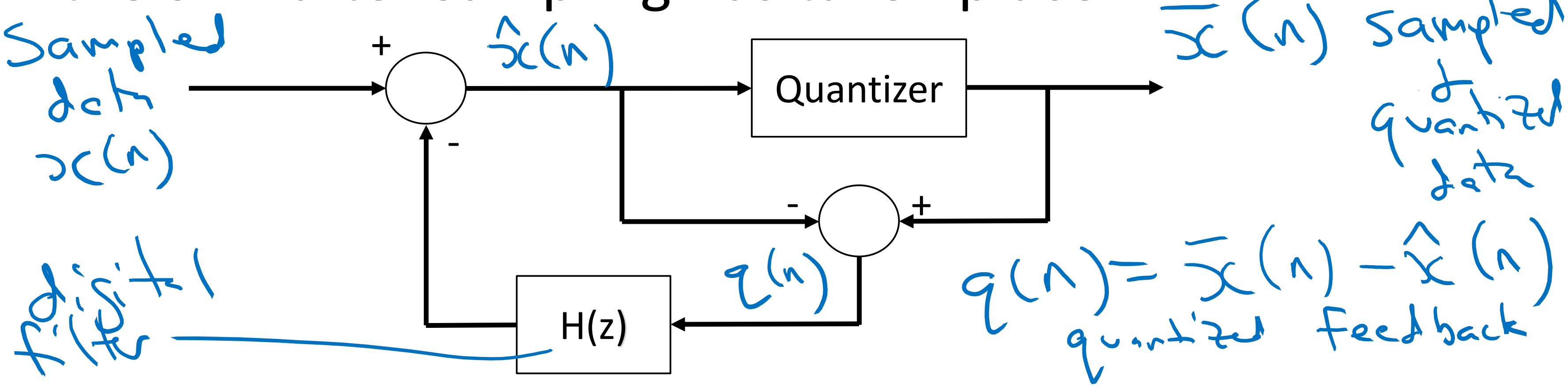
- The signal is still band-limited between 0 and fm. The quantization noise is spread between 0 and 2fm.
- We can use digital filtering to remove the noise between fm and 2fm.

$10 \log_{10}(0.5) = 3\text{dB}$

- The power of the signal stays the same, and we have reduced the power of the noise, so our SNR increases.
- If the quantization noise were uniformly distributed, this would result in a 3dB SNR gain for every factor 2 factor of oversampling.

Noise Shaping

- In reality the error is correlated in time, and the spectrum of noise falls off with frequency. Thus the SNR gain is reduced for each increase in f_s.
- We can use noise shaping within the ADC to improve the SNR after sampling has taken place.



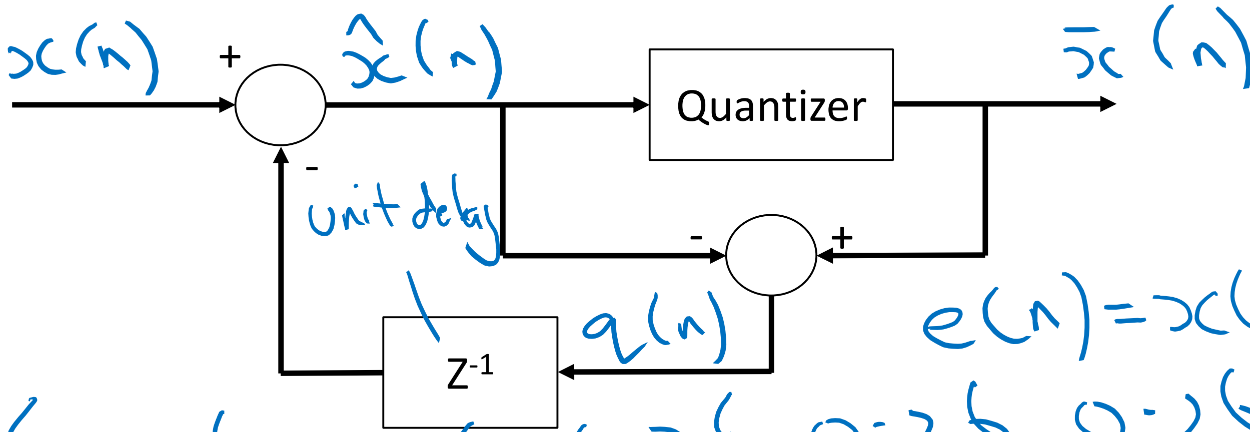
- We feed the quantized error back through a digital filter H(z) (eg delay) which pushes the quantization noise to higher frequencies.

if no feedback $\bar{x} = \{0.3, 0.3, 0.3, 0.3 \ldots\}$ ie DC

$e(n) = \{-0.04, -0.04, -0.04 \ldots\}$

Noise Shaping Example

- Consider quantizing with a step of 0.1 the following DC signal and a unit delay filter for H(z)



$e(n) = x(n) - \bar{x}(n)$

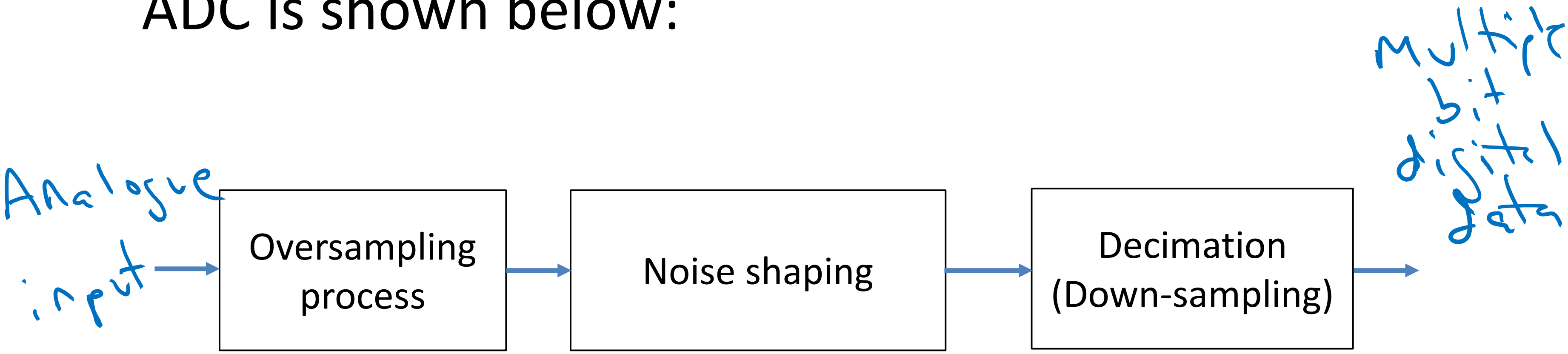
$x(n) = \{0.26, 0.26, 0.26, 0.26, 0.26 \ldots\}$

n	$x(n)$	$\hat{x}(n)$	$\bar{x}(n)$	$q(n)$	$e(n)$
0	0.26	0.26	0.3	0.04	-0.04
1	0.26	0.22	0.2	-0.02	0.06
2	0.26	0.28	0.3	0.02	-0.04
3	0.26	0.24	0.2	-0.04	0.06
4	0.26	0.30	0.3	0	-0.04

- Quantization error is no longer DC but at higher frequencies.

Oversampling & Noise Shaping

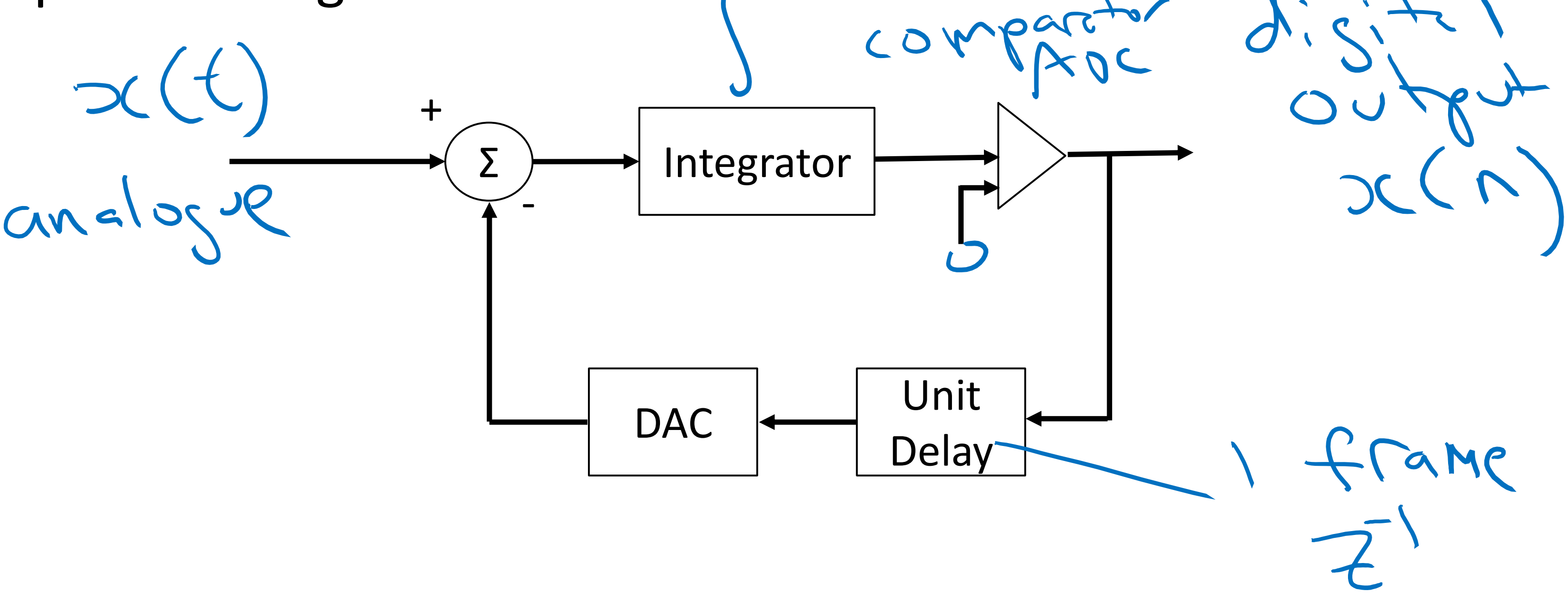
- Noise shaping does not remove the quantization noise, but reshapes it into frequencies that can be digitally filtered.
- The overall process for an oversampled (single-bit) ADC is shown below:



Σ - Δ

Sigma-Delta Converters

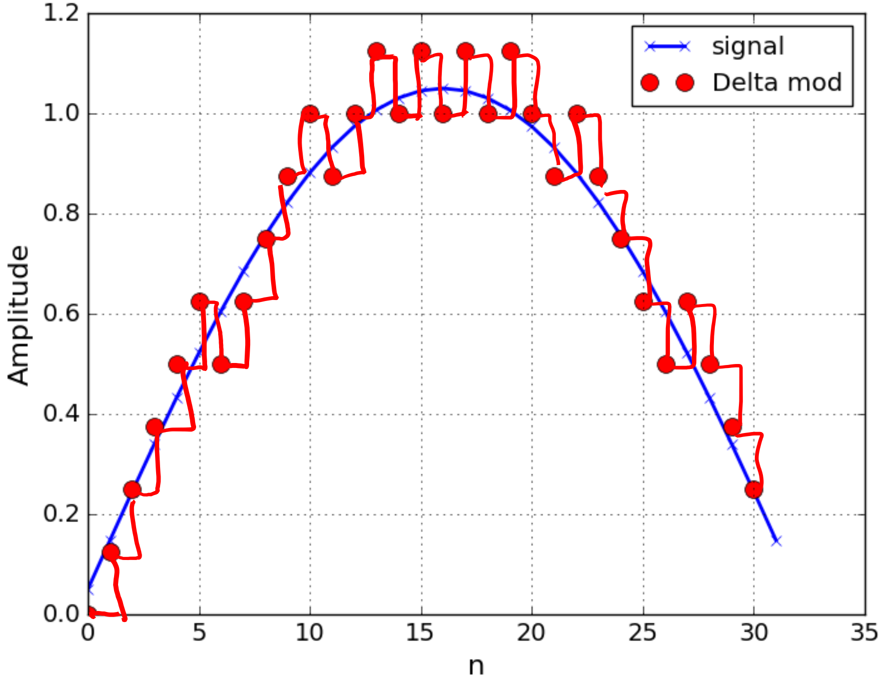
- Σ-Δ conversion is an effective way of achieving noise shaping.
- At each sample the quantized signal is changed by ±Δ depending on whether the last estimate of the quantized signal is above or below the current signal level.
- Output is a series of deltas that we integrate to get the quantized signal.



Σ-Δ convertor example

- At each sample the quantized signal is changed by ±Δ depending on whether the last estimate of the quantized signal is above or below the current signal level.
- Output is a series of deltas that we integrate to get the quantized signal.

$\Delta = 0.125$

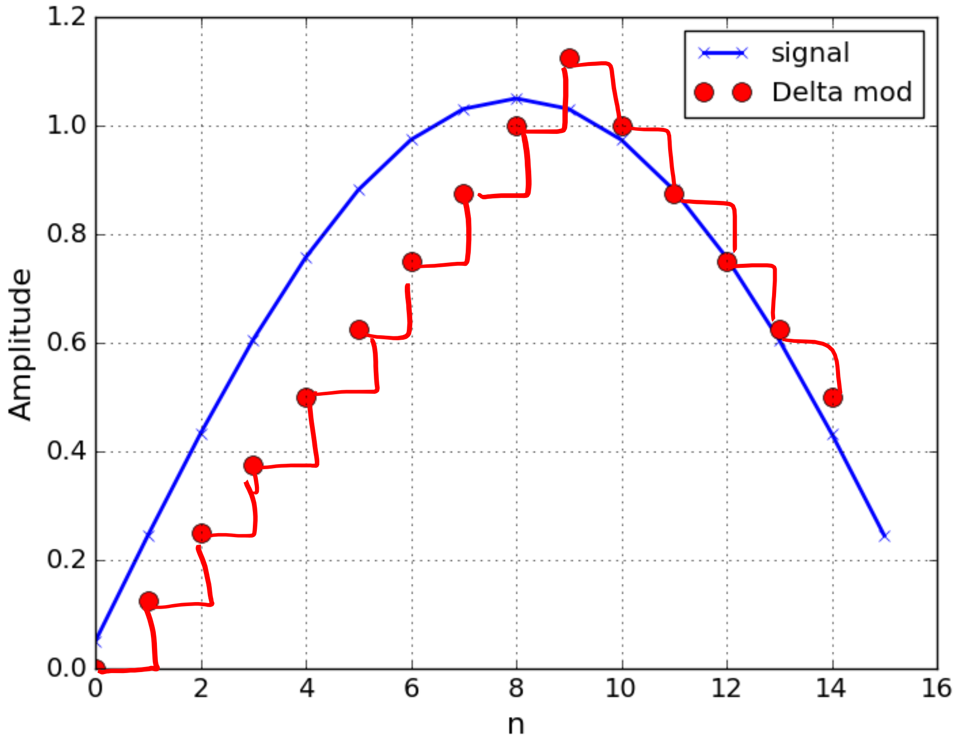


Slope overload

$\Rightarrow$ Can't keep up

- If the input signal changes too fast (f_s not fast enough), we can end up with slope overload distortion.

$\Delta = 0.125$

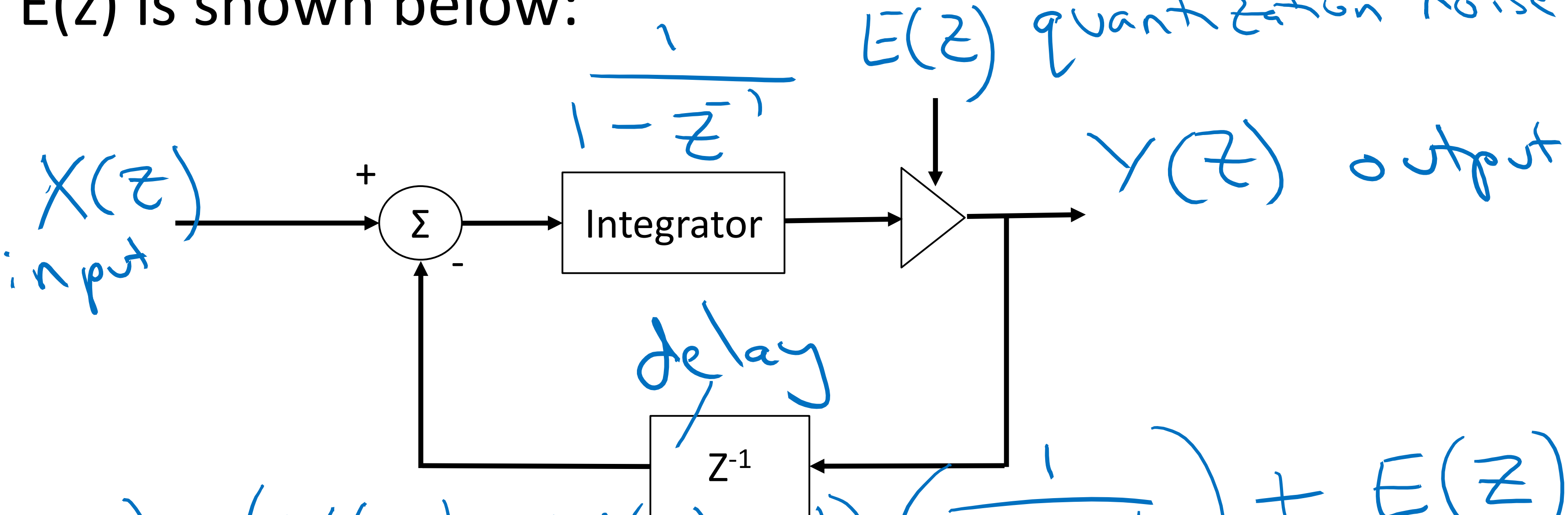


Halved f_s compared to previous slide.

$(1 - z^{-1})$ is differentiator $\Rightarrow$ noise shaping

Z-transform model of Σ-Δ convertor

- The Z-transform model of the Σ-Δ convertor, where the quantization noise is injected as an error signal E(z) is shown below:



$$Y(z) = \left(X(z) - Y(z)z^{-1}\right)\left(\frac{1}{1-z^{-1}}\right) + E(z)$$

$$Y(z)(1-z^{-1}) = X(z) - Y(z)z^{-1} + E(z)(1-z^{-1})$$

$$Y(z) = X(z) + E(z)(1-z^{-1})$$

Z-transform model of Σ-Δ convertor 2

- The differentiator acts as a high pass filter with zero at DC. It acts to push the quantization noise energy up to the higher frequencies.

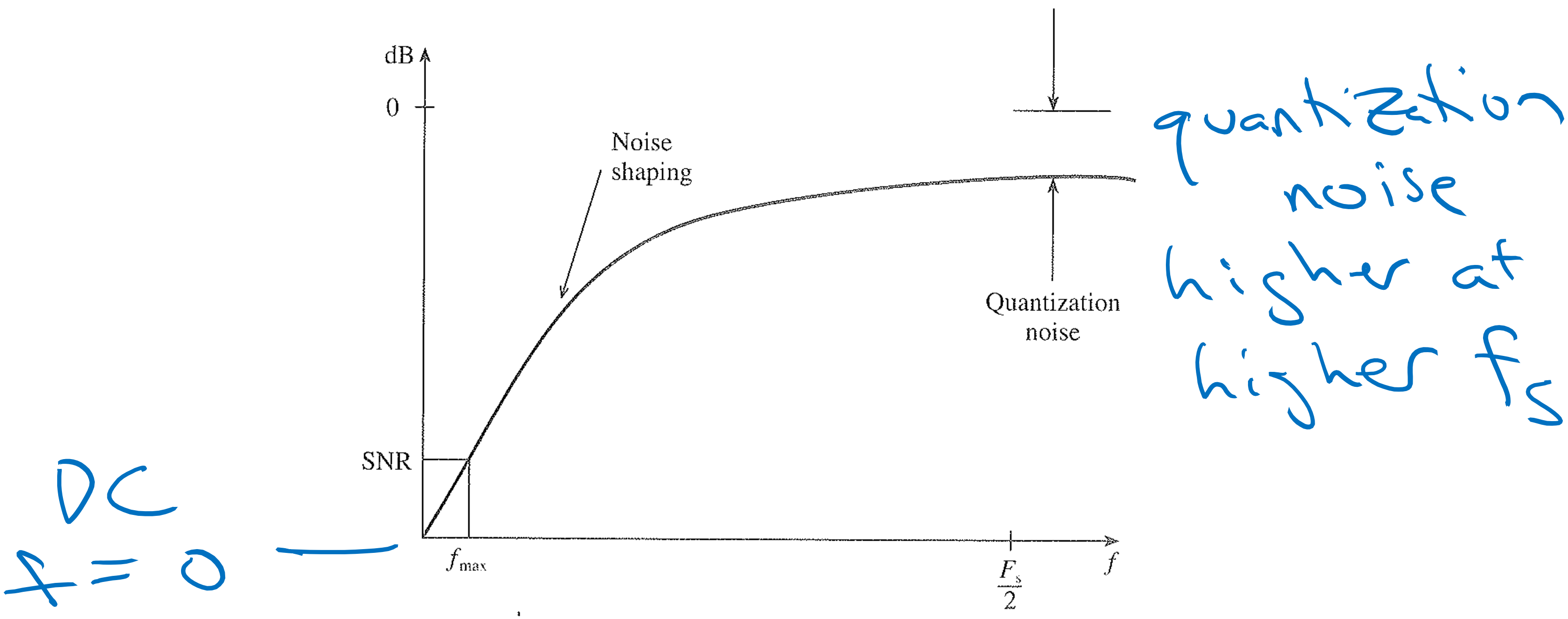


Figure 2.38 Effects of noise shaping on quantization noise.

From I & J

Second Order Σ-Δ convertor

- First stage is the same as the first order Σ-Δ convertor. The input to the second stage is the quantization noise $E_1(Z)$.

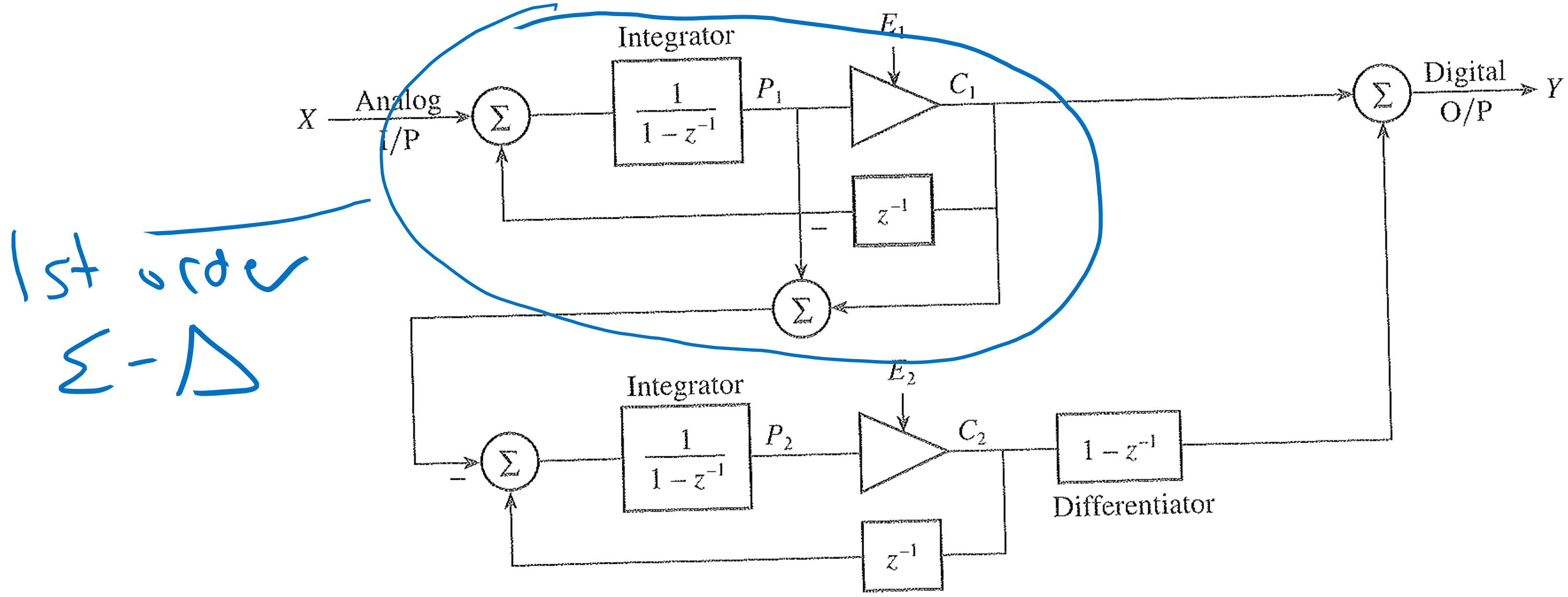


Figure 2.37 *z*-plane model of second-order SDM (Example 2.12).

I ∂ ∫

$$Y(z) = X(z) + E_2(z)(1-z^{-1})^2$$