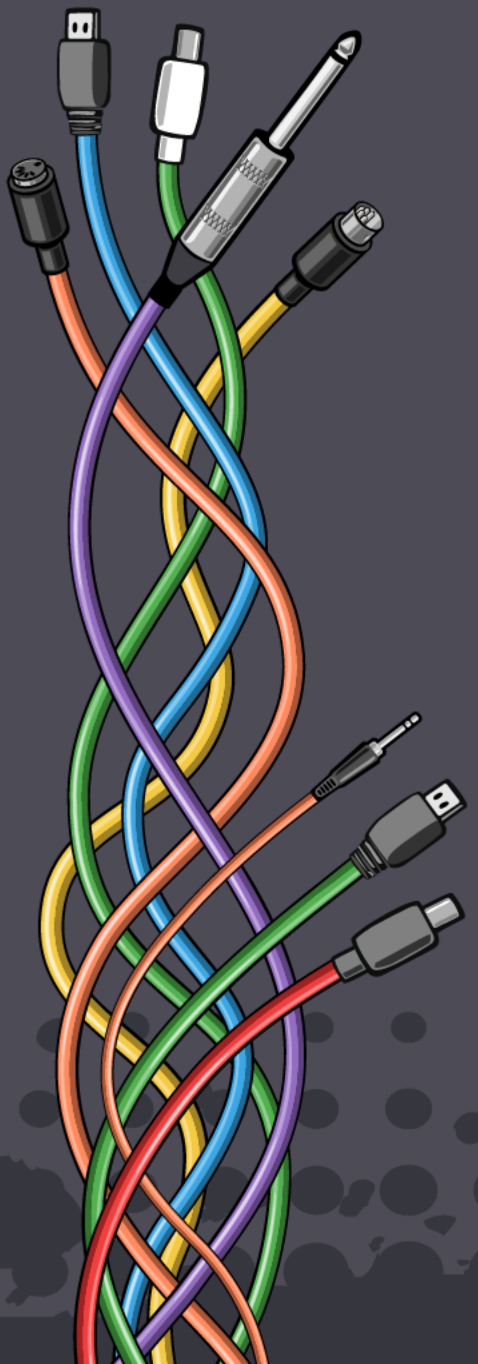
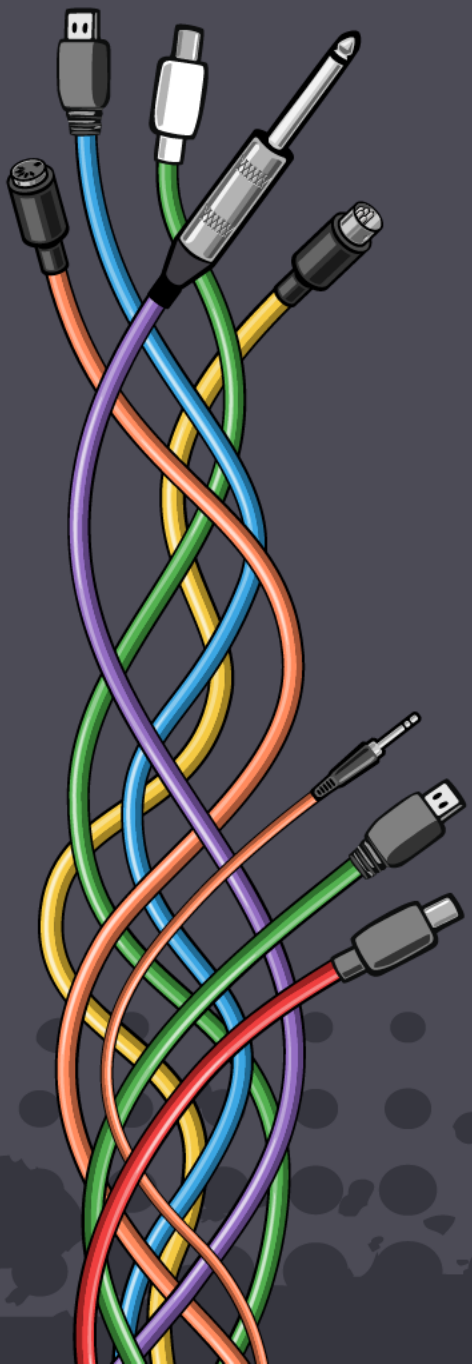


# THE REAL WAVEFORM MATTERS

*THE SAMPLES ARE NOT ALWAYS WHAT THEY SEEM*

**JAMIE ANGUS-WHITEOAK**



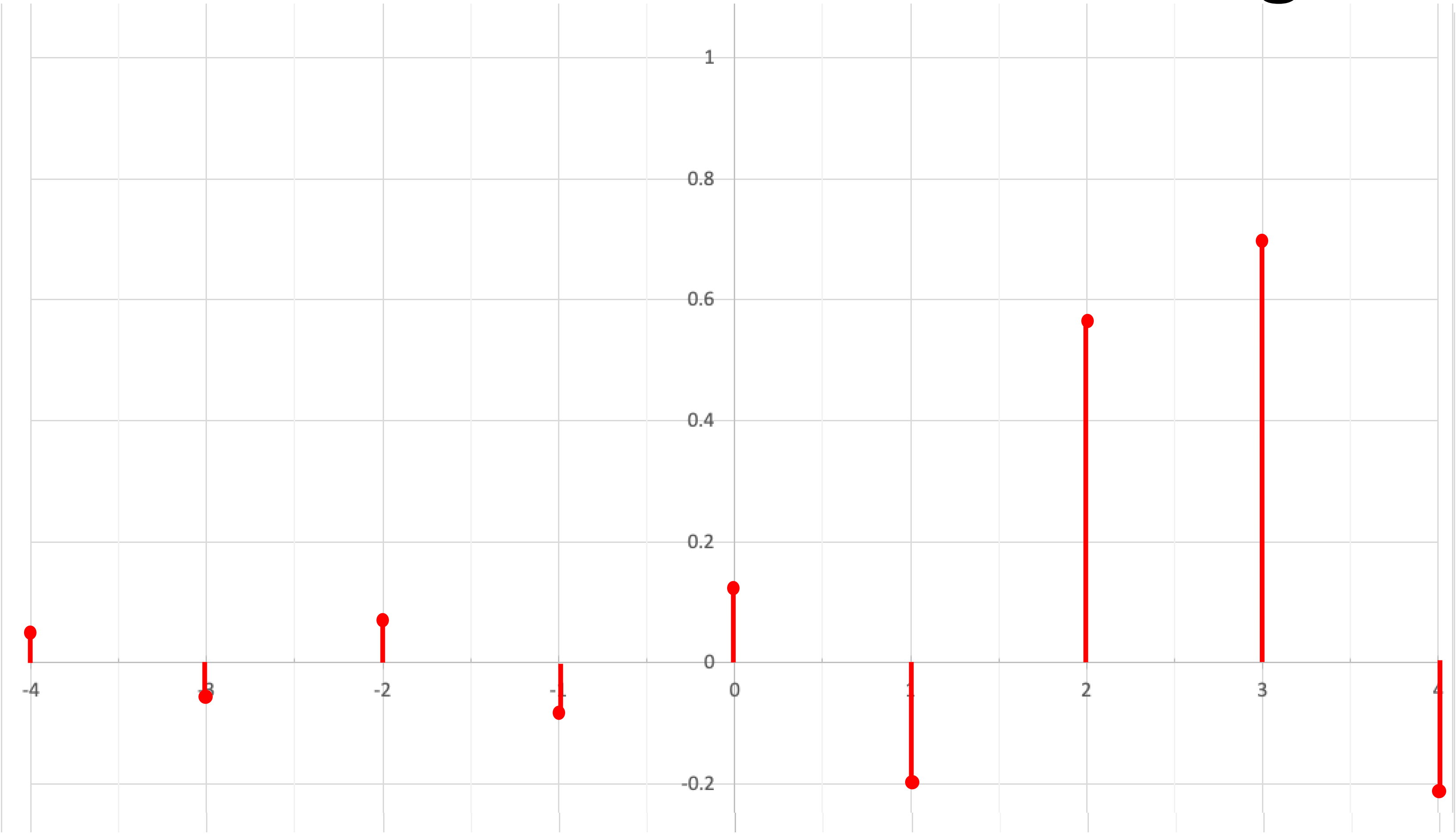
# THE REAL WAVEFORM MATTERS

*THE SAMPLES ARE NOT ALWAYS WHAT THEY SEEM*

**JAMIE ANGUS-WHITEOAK**

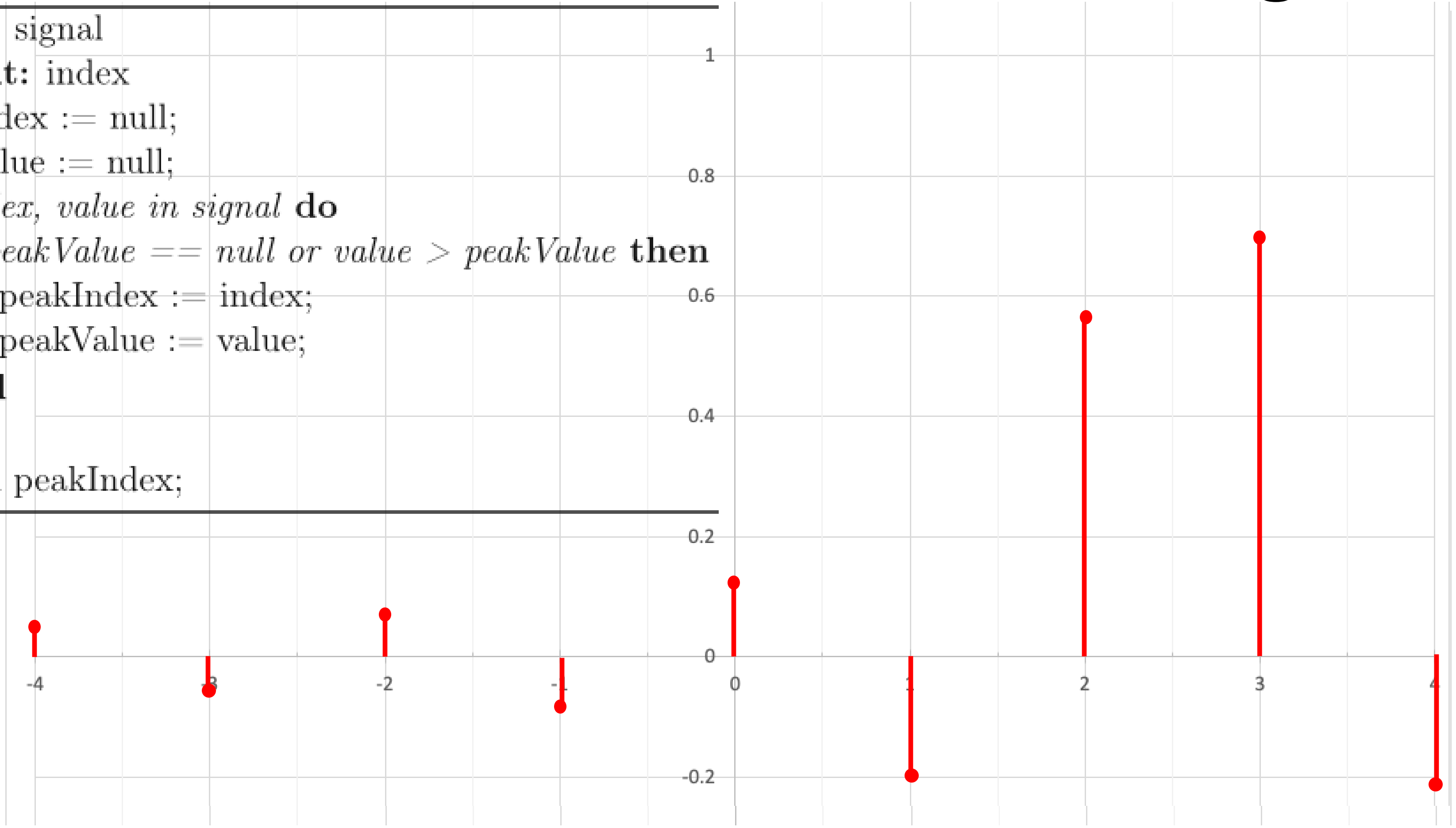
ADC25 Bristol

# What Is The Peak Value Of This Signal?

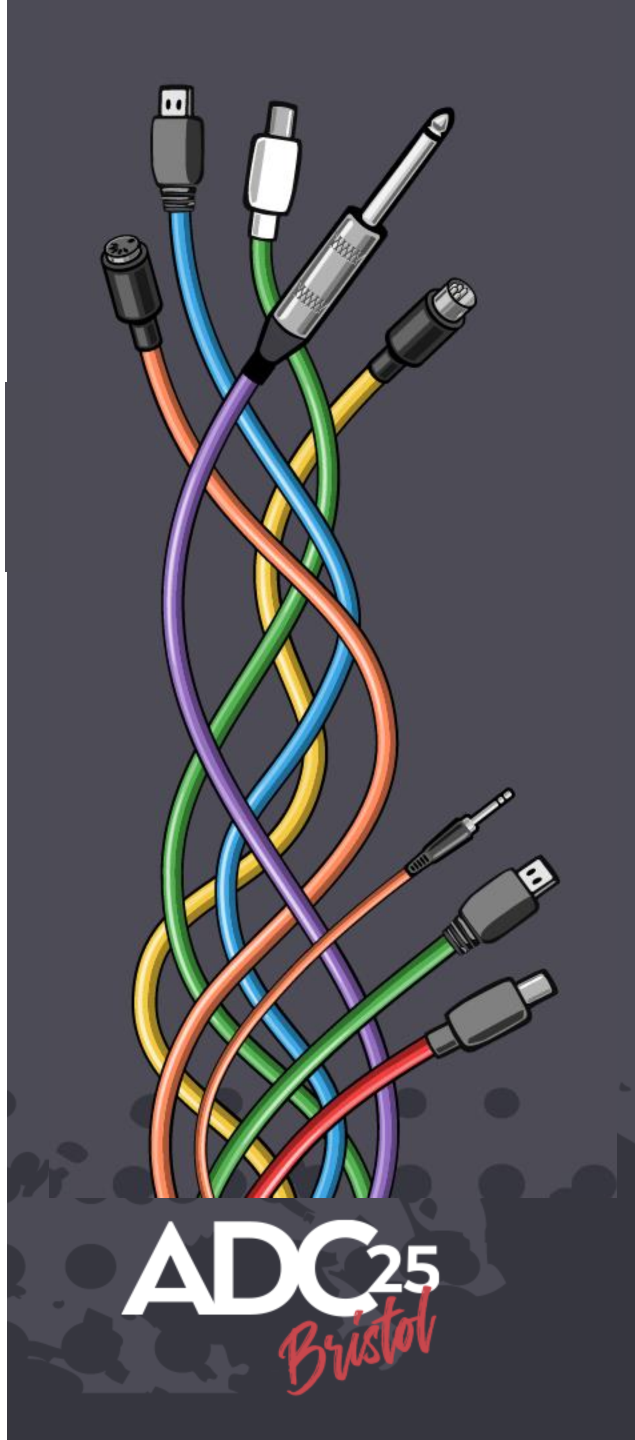


# What Is The Peak Value Of This Signal?

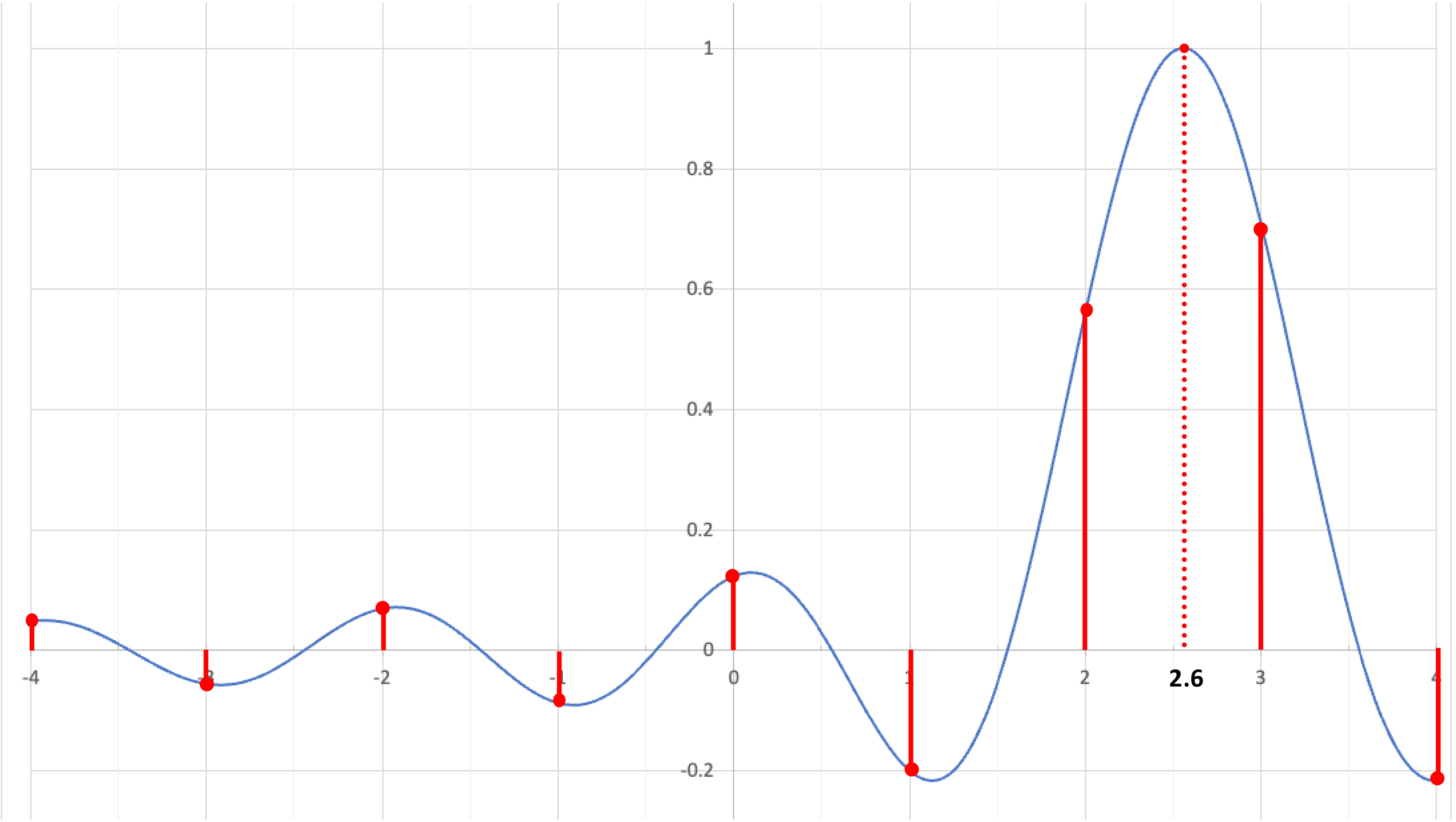
```
Input: signal
Output: index
peakIndex := null;
peakValue := null;
for index, value in signal do
    if peakValue == null or value > peakValue then
        peakIndex := index;
        peakValue := value;
    end
end
return peakIndex;
```



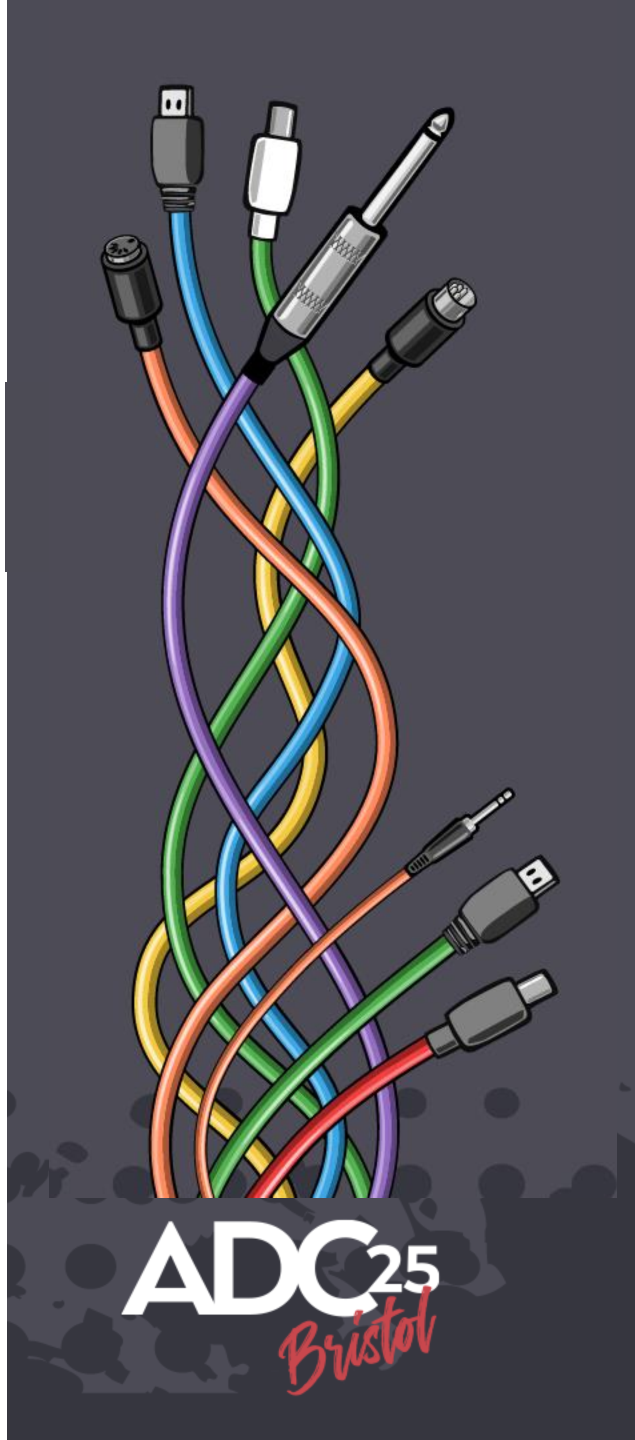

- A simplistic examination of sample values would suggest:
  - Peak sample value equal to 0.7
  - At sample number +3?



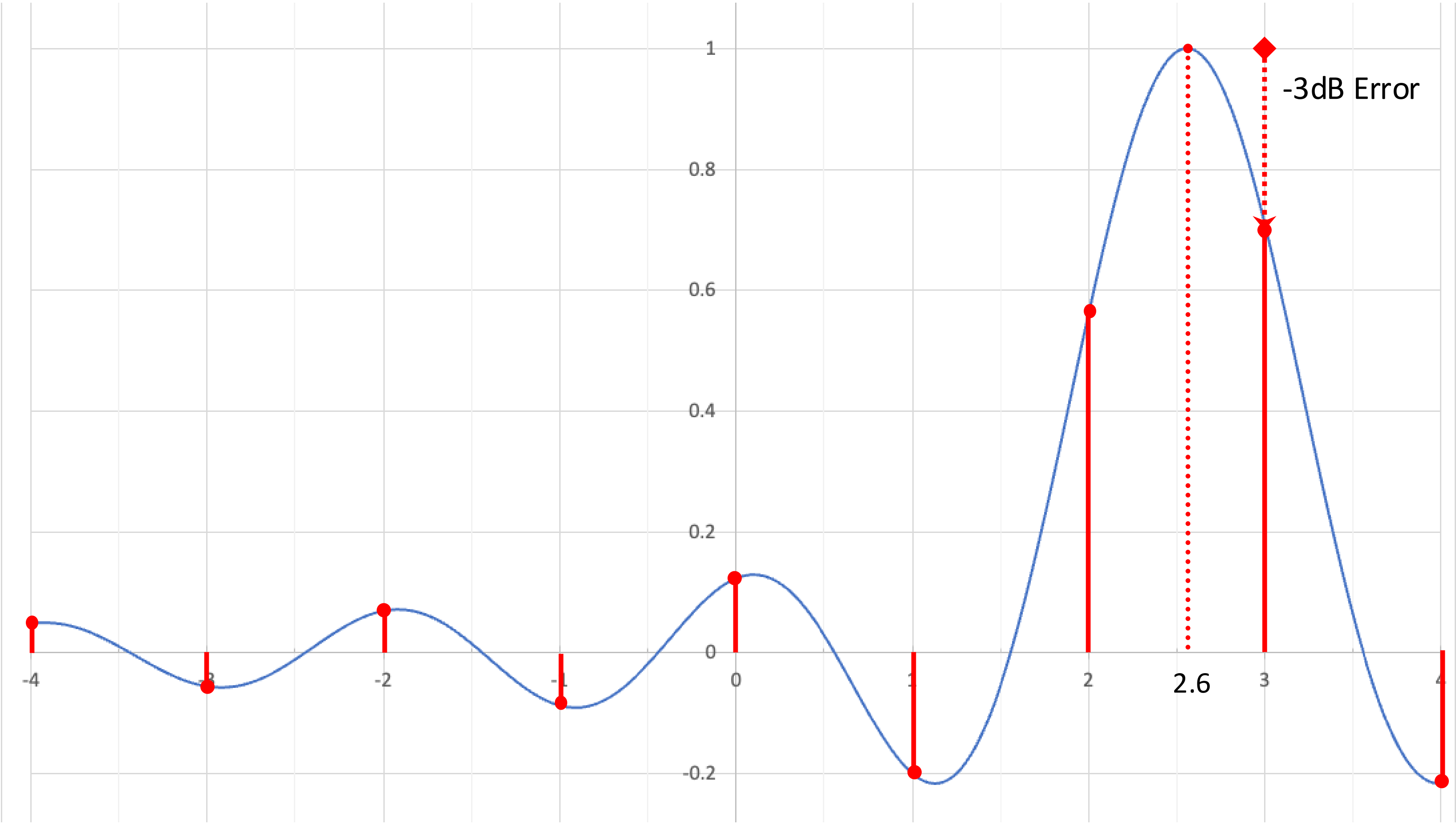
# What Is The Peak Value Of This Signal?



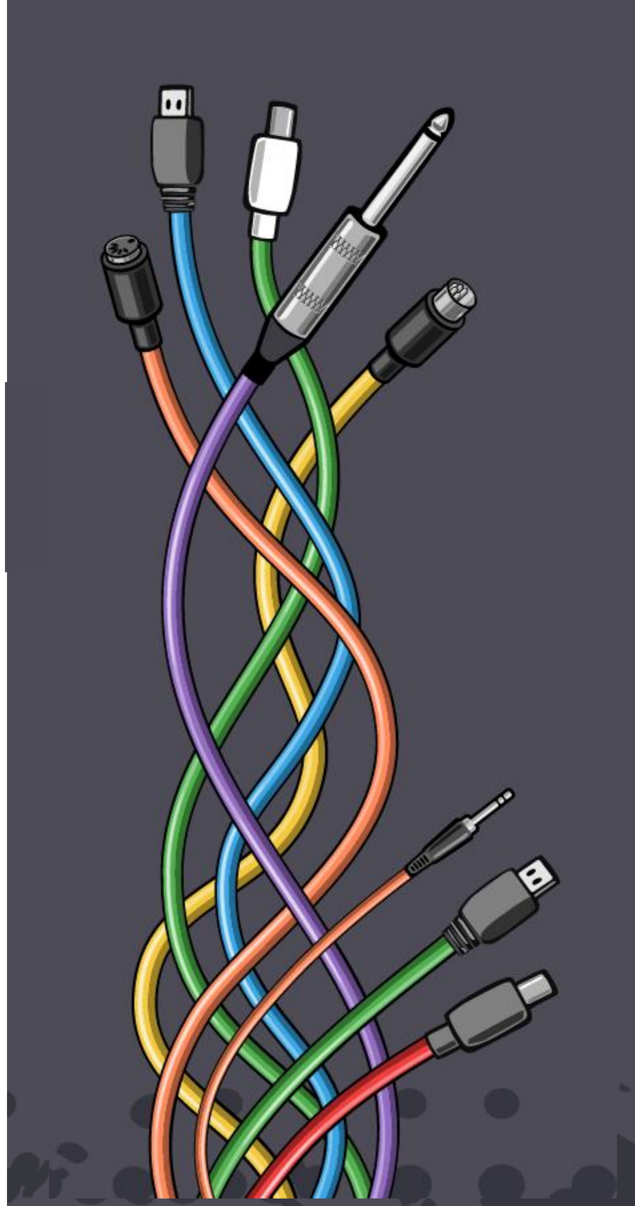

- But you would be wrong!
  - The peak value is actually 1!
  - And occurs between two samples at position 2.6



# What Is The Peak Value Error?

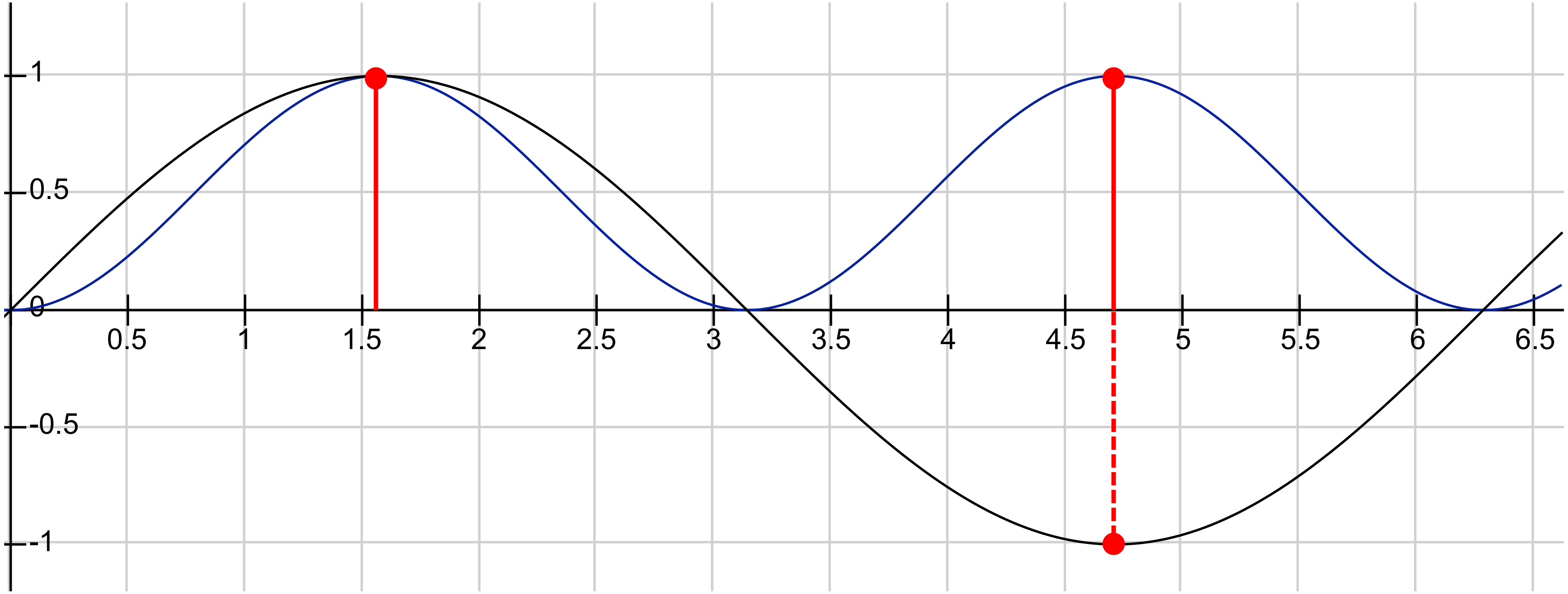



- This is a level error of -3dB!
  - That's significant in many applications
  - The time shift may also matter



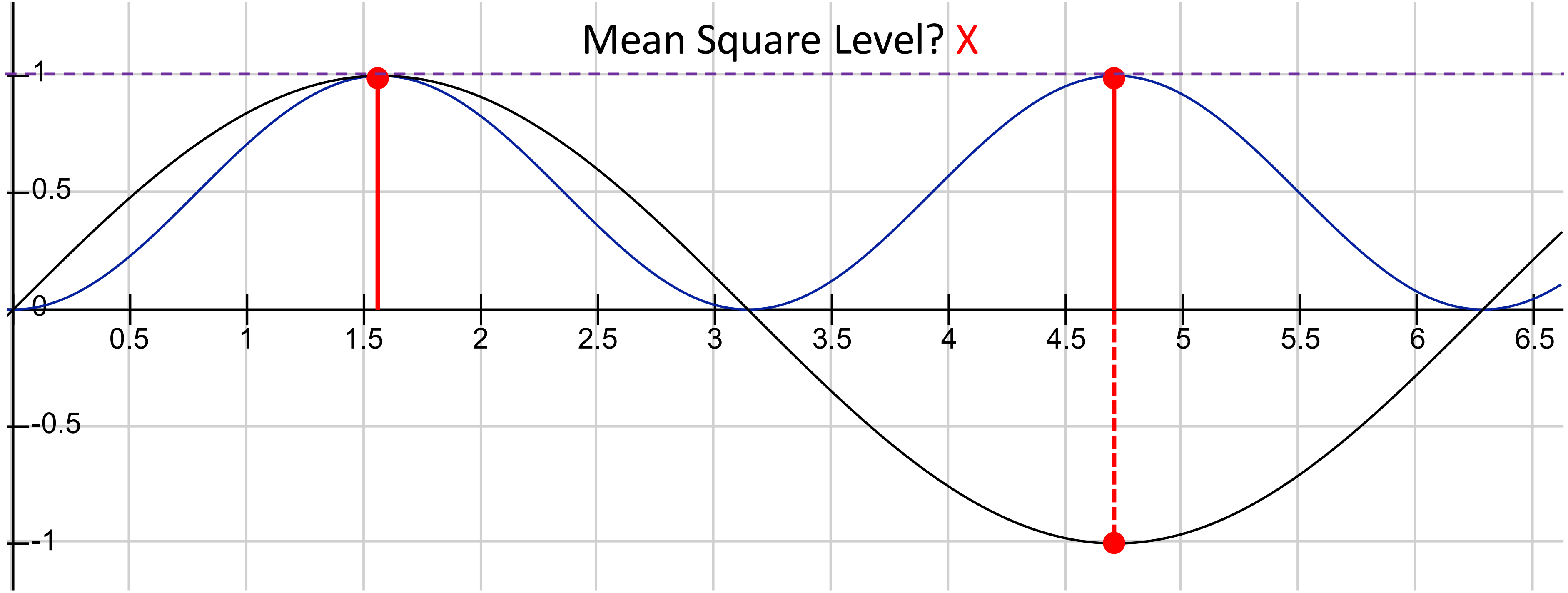
ADC25 Bristol

# Non-Linear Operations

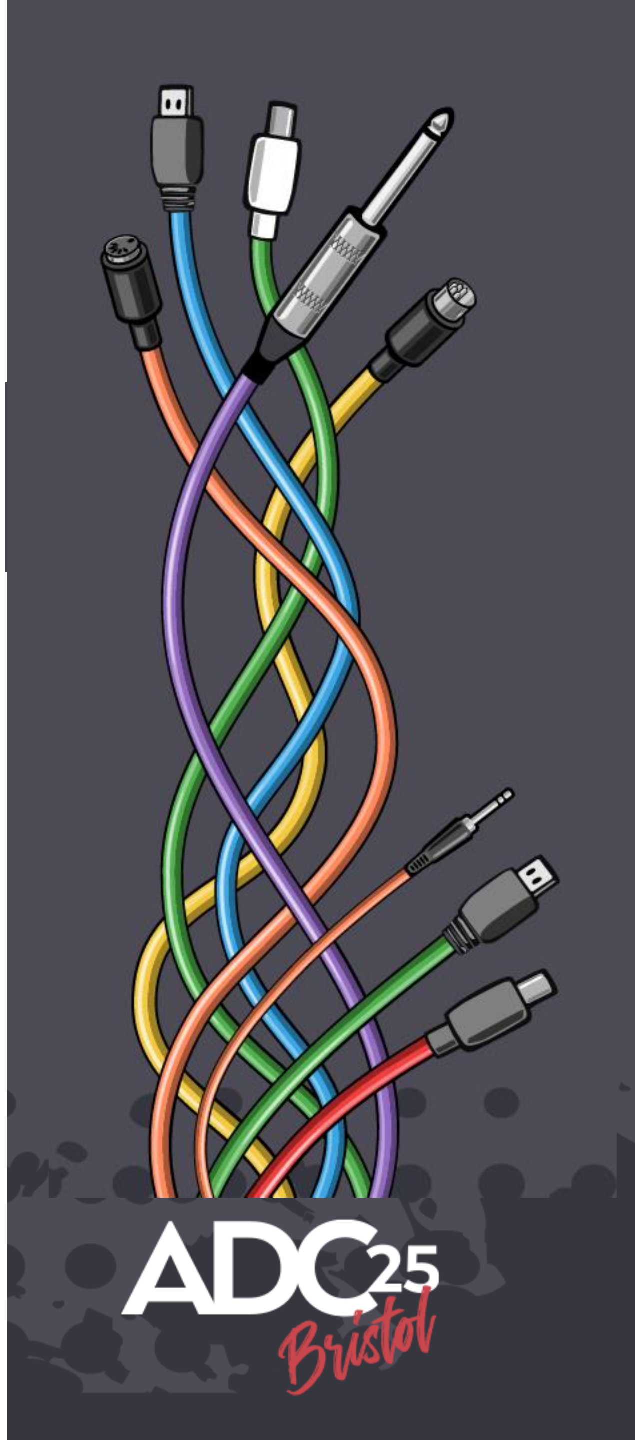



- If nonlinear operations are applied to samples e.g. ($x^2$)
- It means that parts of the waveform are missed
  - Between the samples

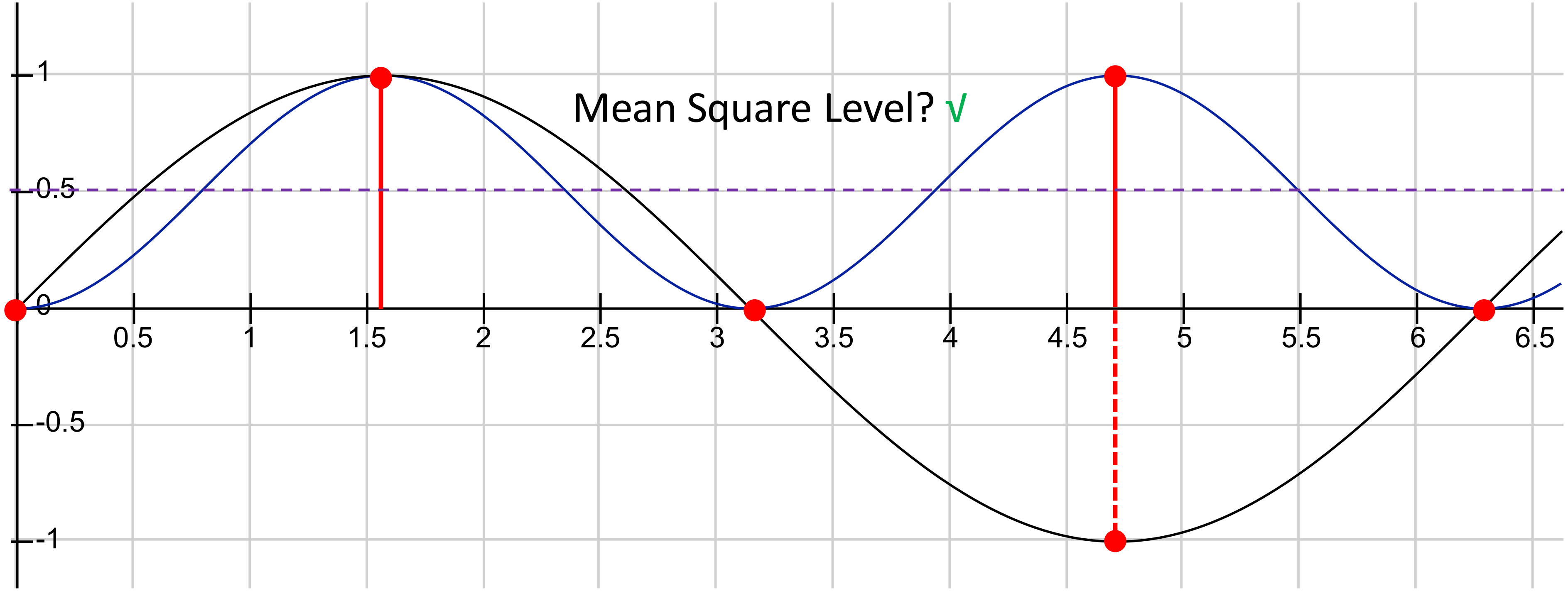
# Non-Linear Operations



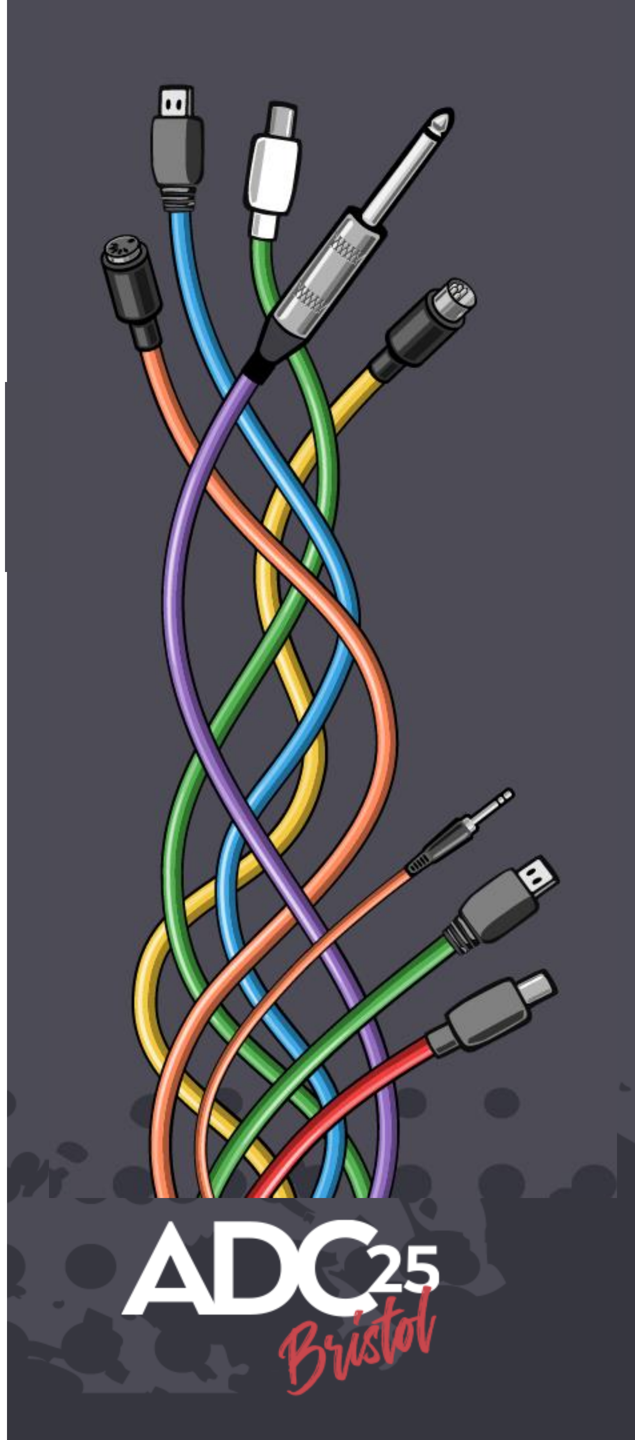

- If nonlinear operations are applied to samples e.g. ($x^2$)
- It means that parts of the waveform are missed
  - Between the samples
  - For example, the zeros in this case
  - This means even RMS won't work properly!



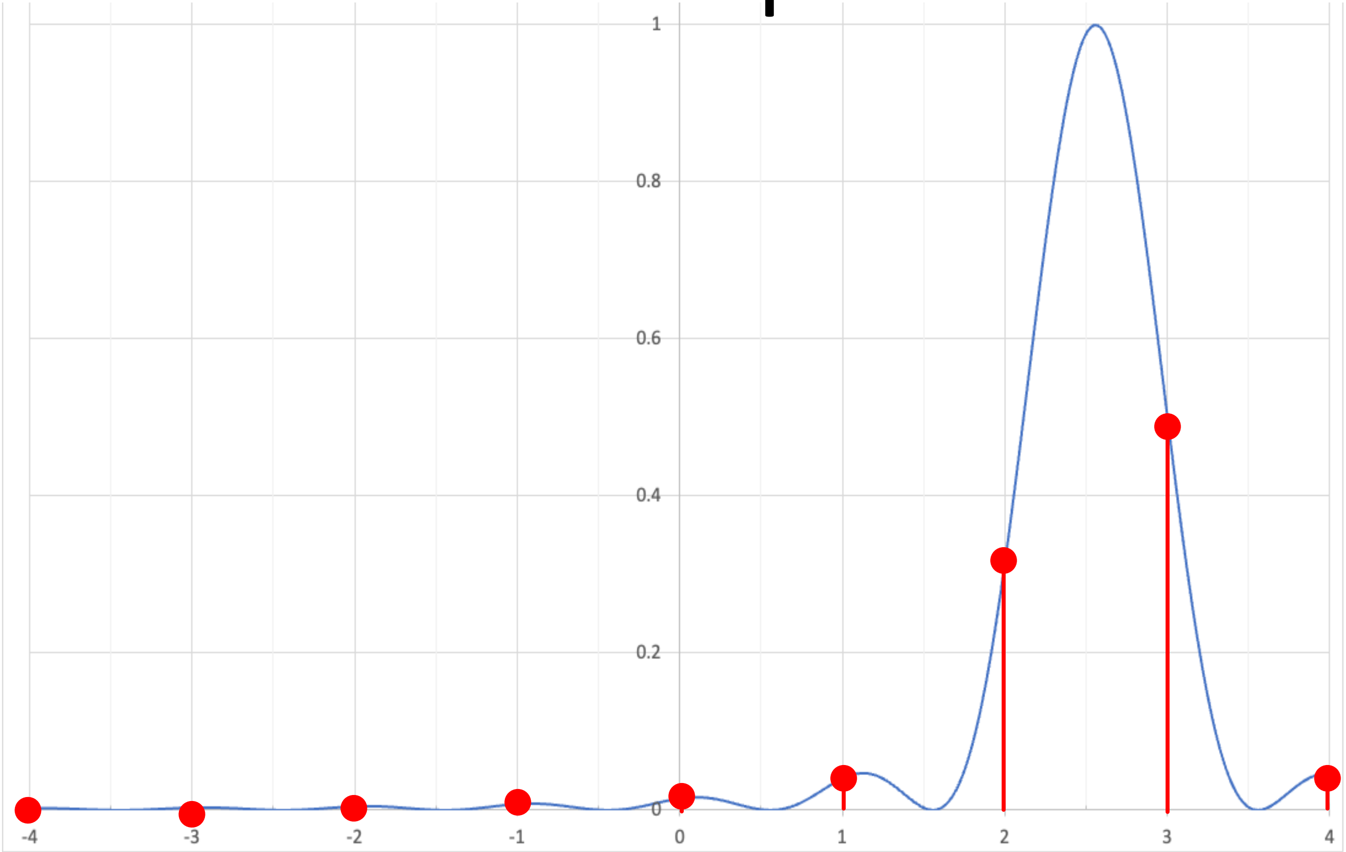
# Non-Linear Operations



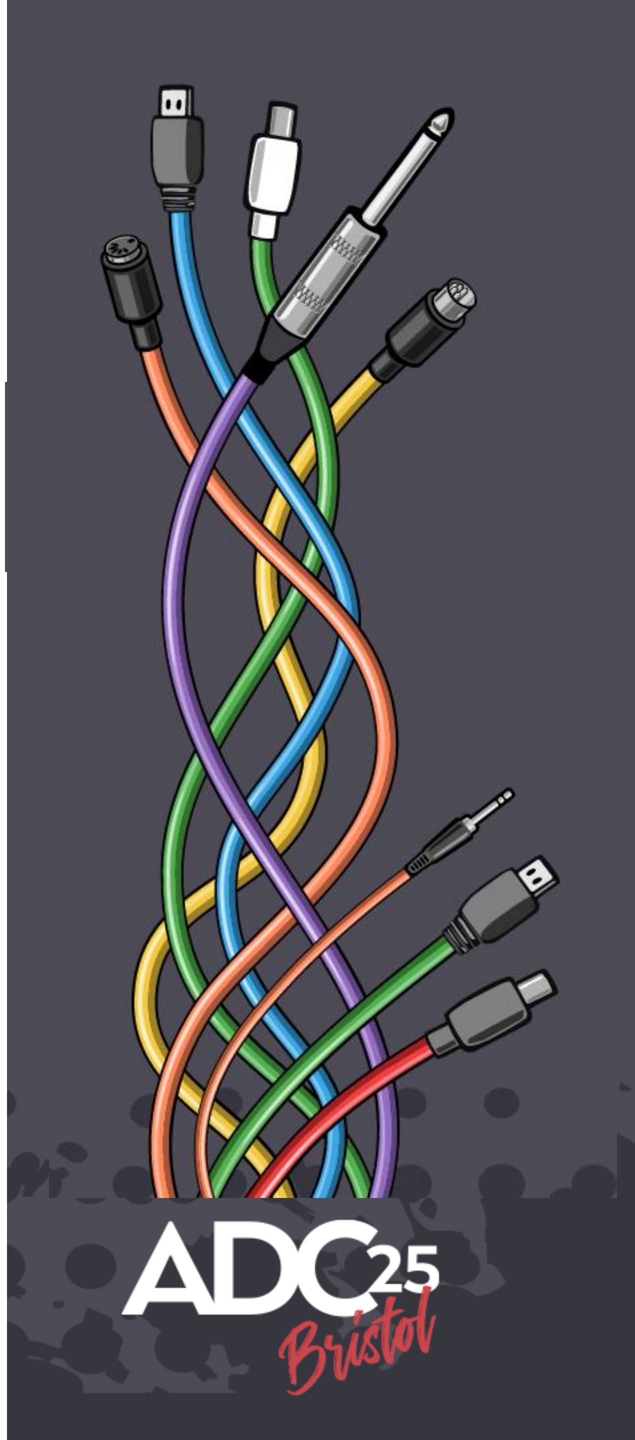

- If nonlinear operations are applied to samples e.g. ($x^2$)
- It means that parts of the waveform are missed
  - Between the samples
- We need to oversample by at least factor of 2 to get the correct RMS for $F_s/2$



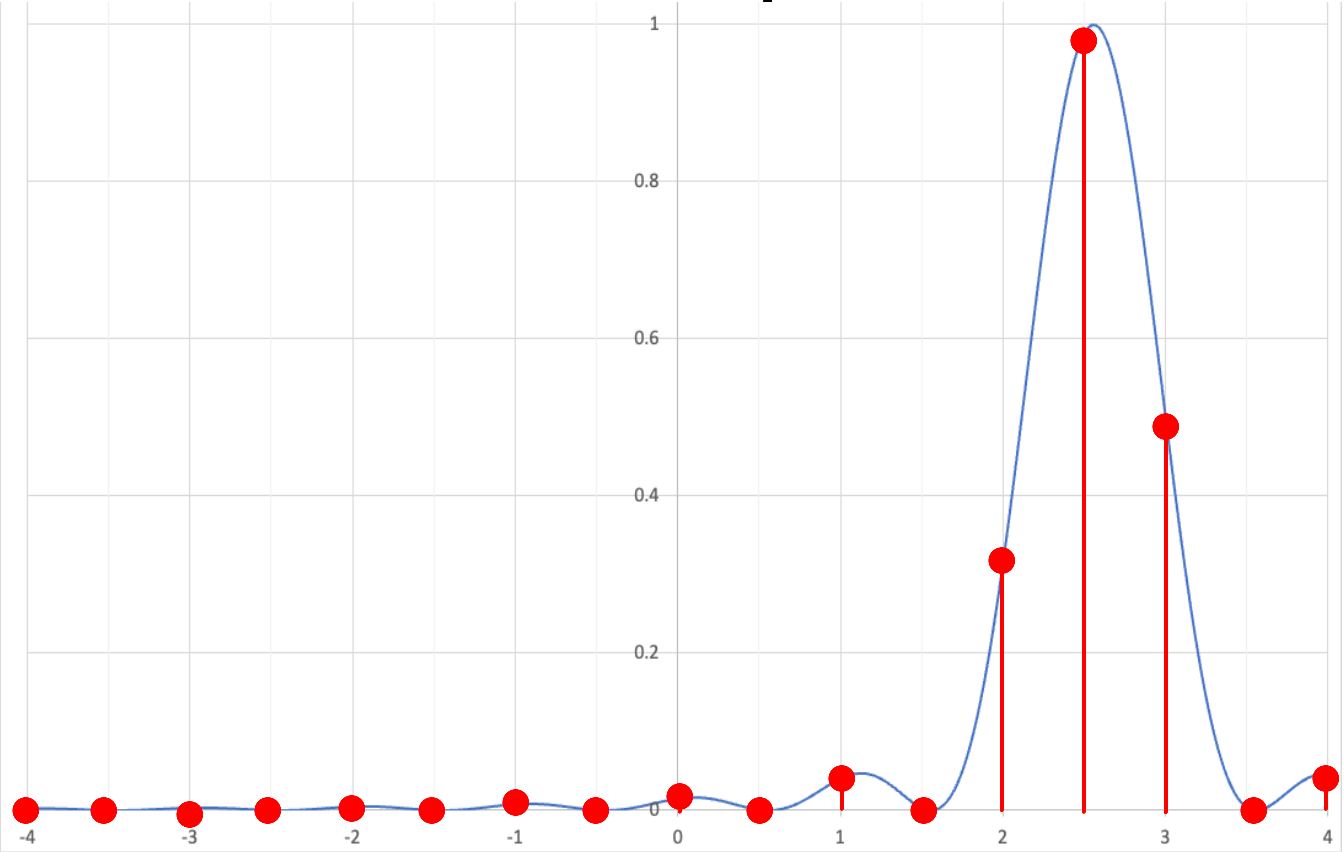
# Non-Linear Operations



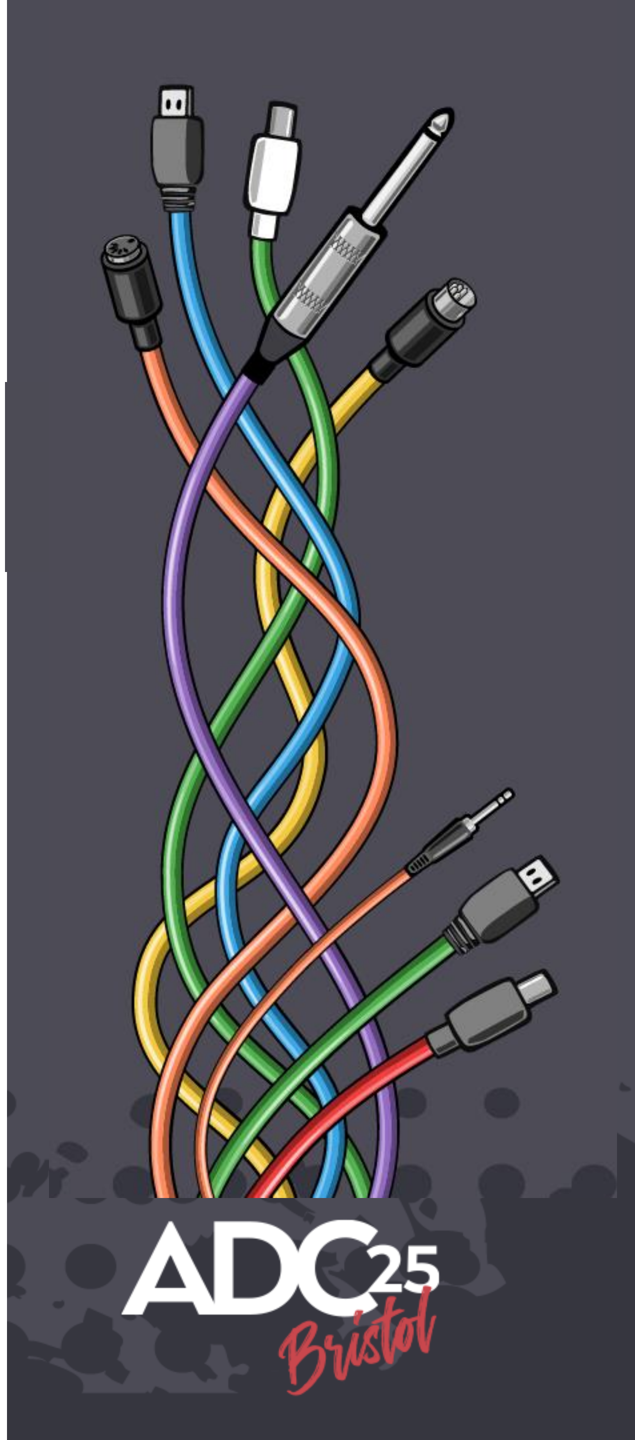

- If nonlinear operations are applied to samples e.g. ($x^2$)
- It means that parts of the waveform are missed
  - Between the samples
  - For example, the zeros in this case
  - Can be significant when finding the magnitude of a Fourier transform



# Non-Linear Operations

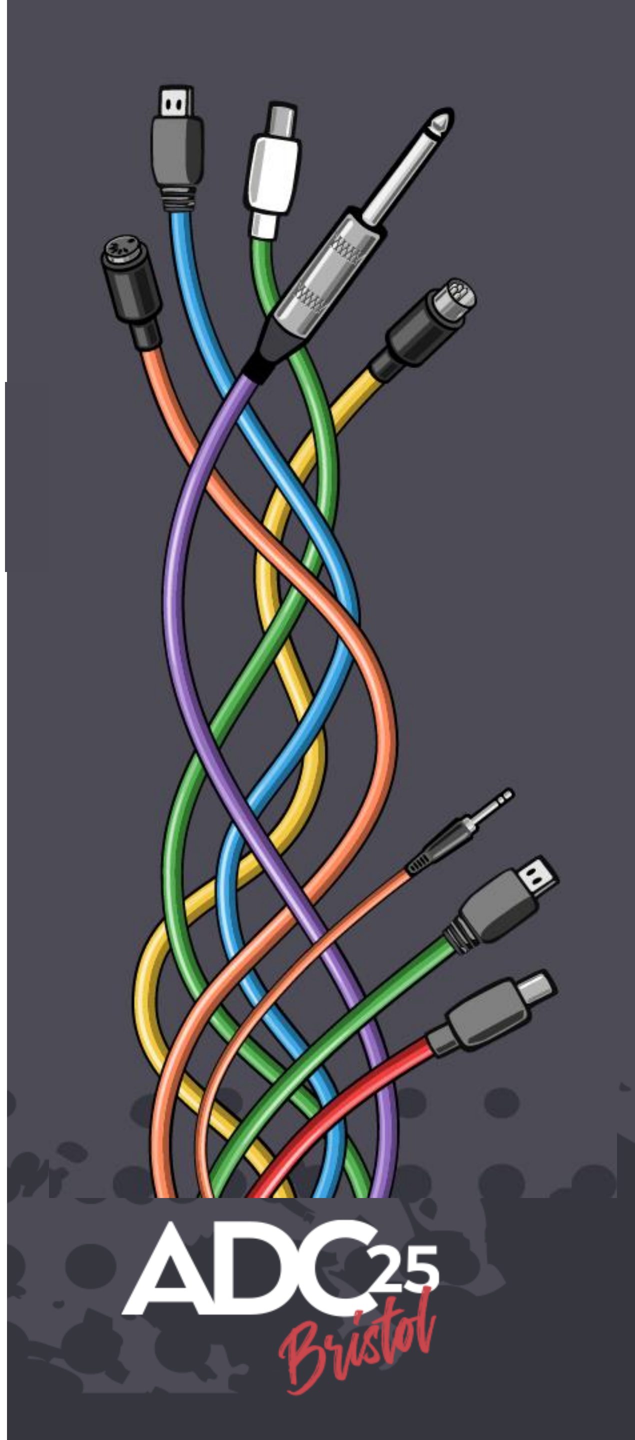



- If nonlinear operations are applied to samples e.g. ($x^2$)
- It means that parts of the waveform are missed
  - Between the frequency samples
  - Can be significant when finding the magnitude of a Fourier transform
  - Need to increase in the frequency resolution via interpolation



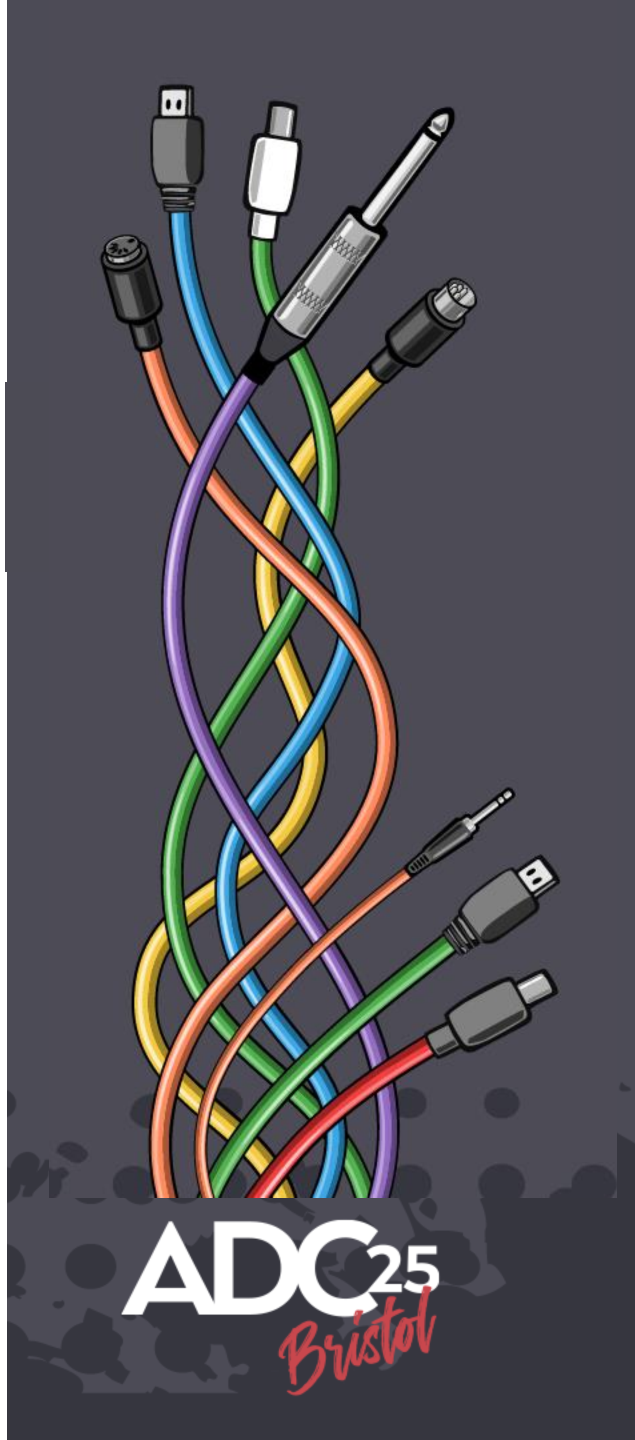
# The Samples Are Not The Whole Story!

- When an audio waveform is sampled
  - The actual waveform level may not be the sample value
- Shannon-Nyquist theory is so pervasive we forget
  - There are terms and conditions attached
  - It assumes that the signal is processed in a linear fashion
  - And that there are no time varying operations
  - That is fixed filters!
- Non-linear operations are not covered
- Neither is time variation
  - Interpolation/decimation
  - Pitch shifting
  - Adaptive filtering

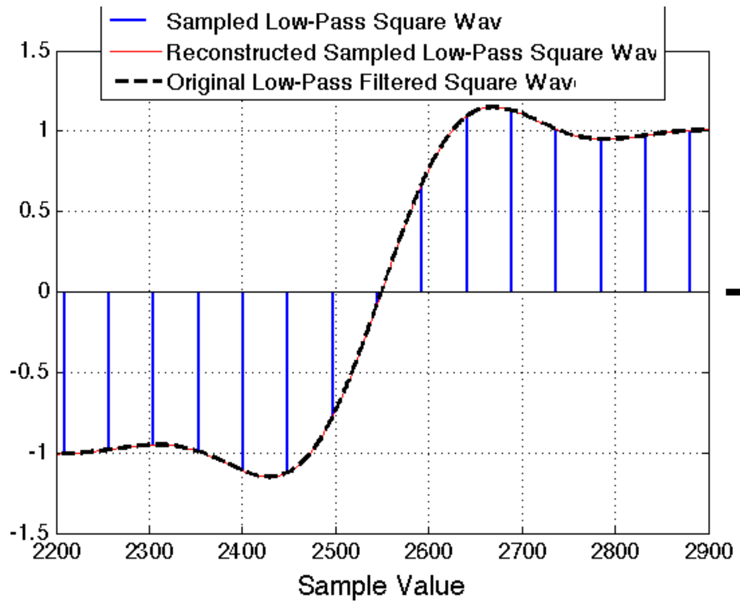


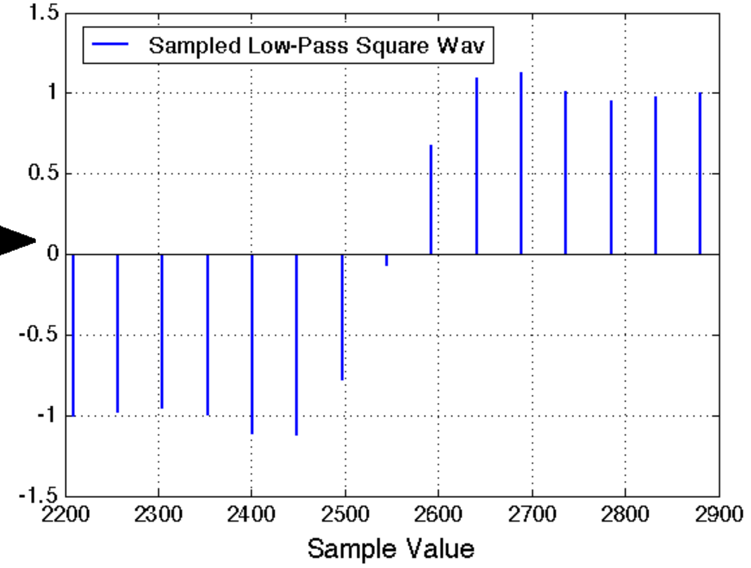
# Structure

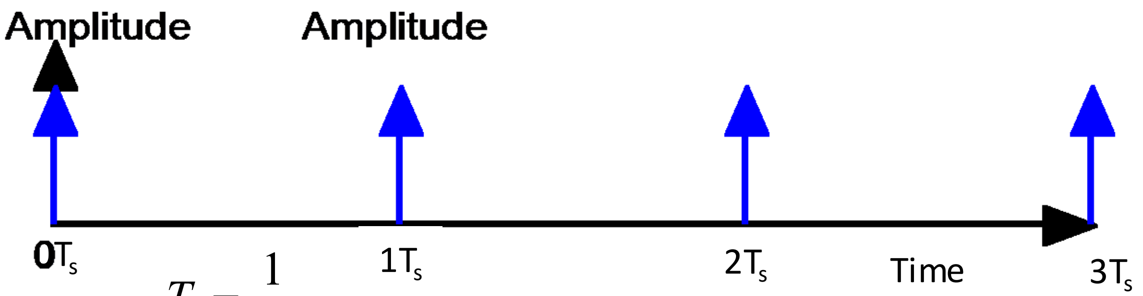
- Why does this happen in sampled signals?
- How do we get the original waveform back?
- An example
  - Where the difference between analogue and sampled waveforms matter
- Oversampling as a solution
- Quadratic interpolation as a solution
- β-Spline interpolation as a solution



# Why Does This Happen? Sampling In Time



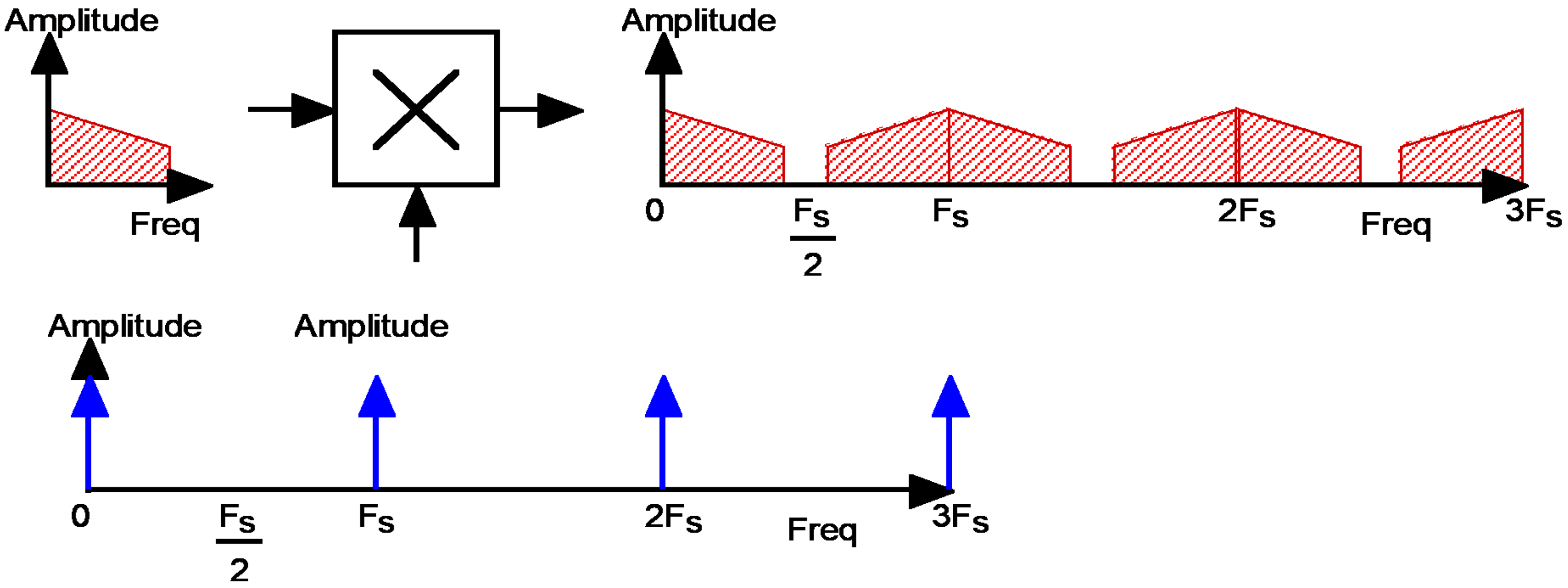



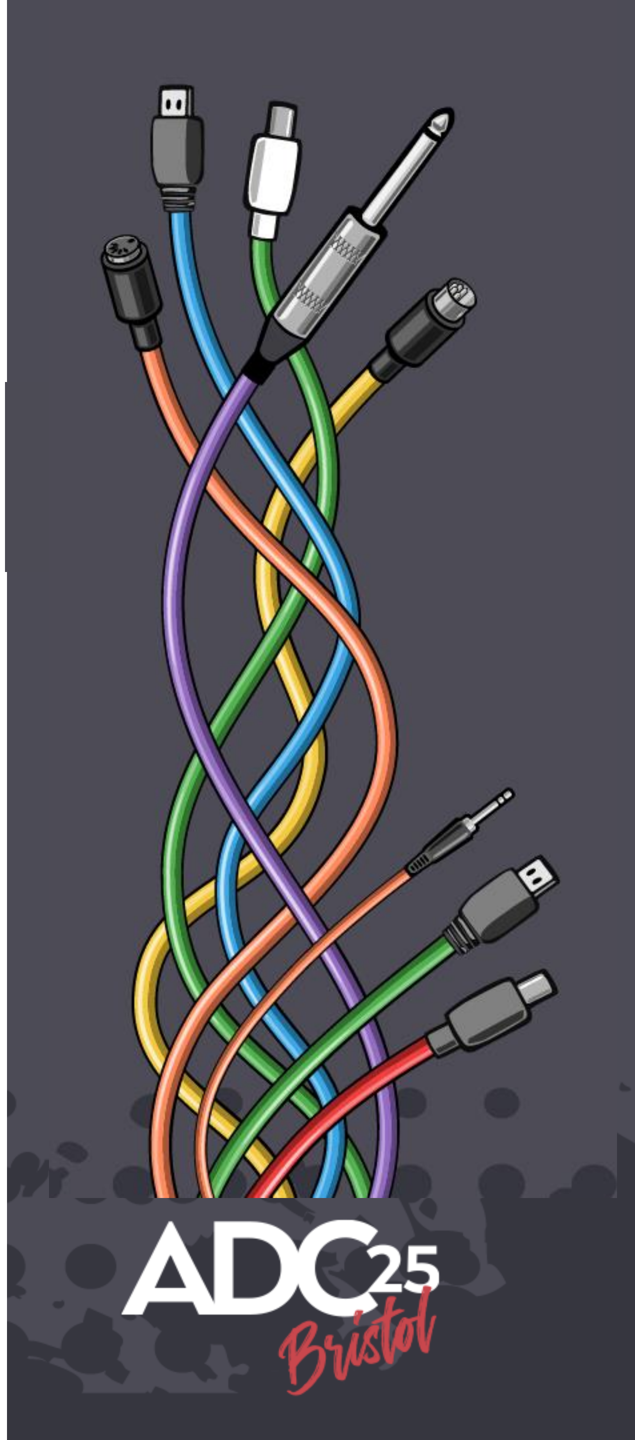

$$T_s = \frac{1}{F_s}$$

- The analogue signal is sampled at discrete time intervals

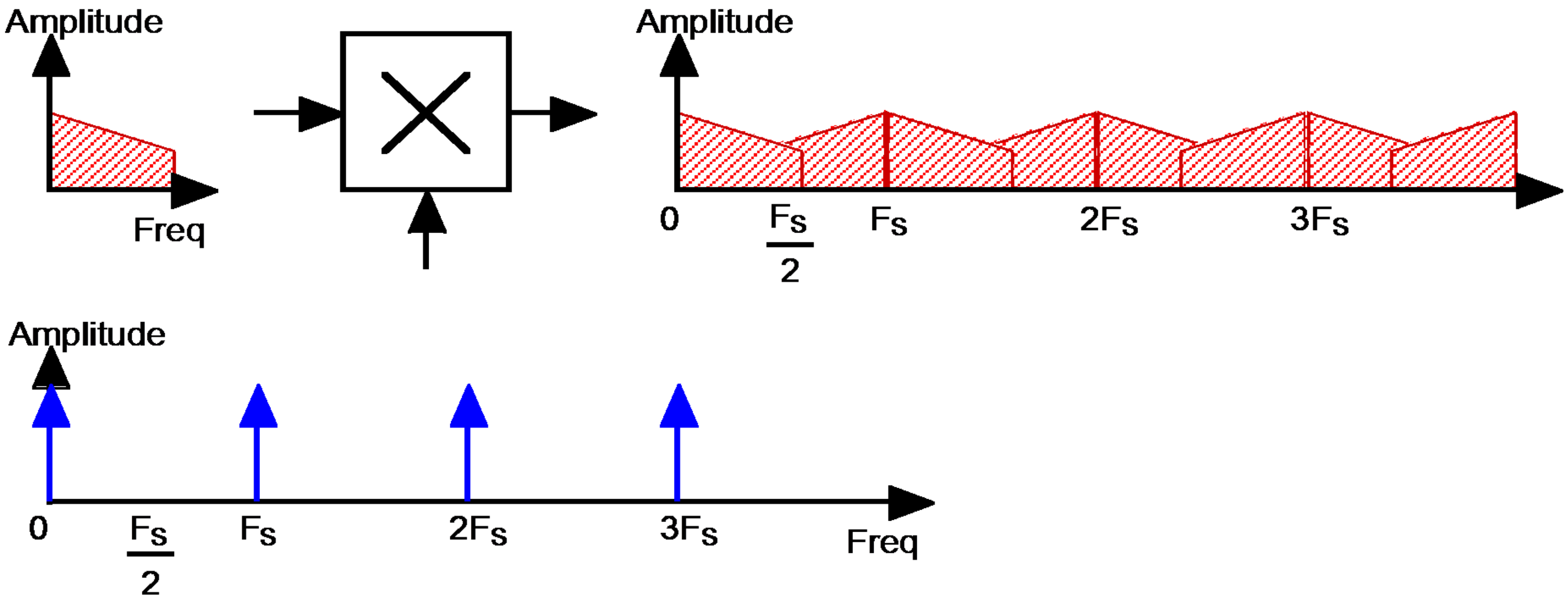
# Sampling In The Frequency Domain




- This is like double sideband supressed carrier modulation
- On an infinite comb of carriers
  - Baseband translated to all these carrier frequencies
  - The infinite numbers of replicants is why the sample amplitudes
  - Can be different from the waveform value

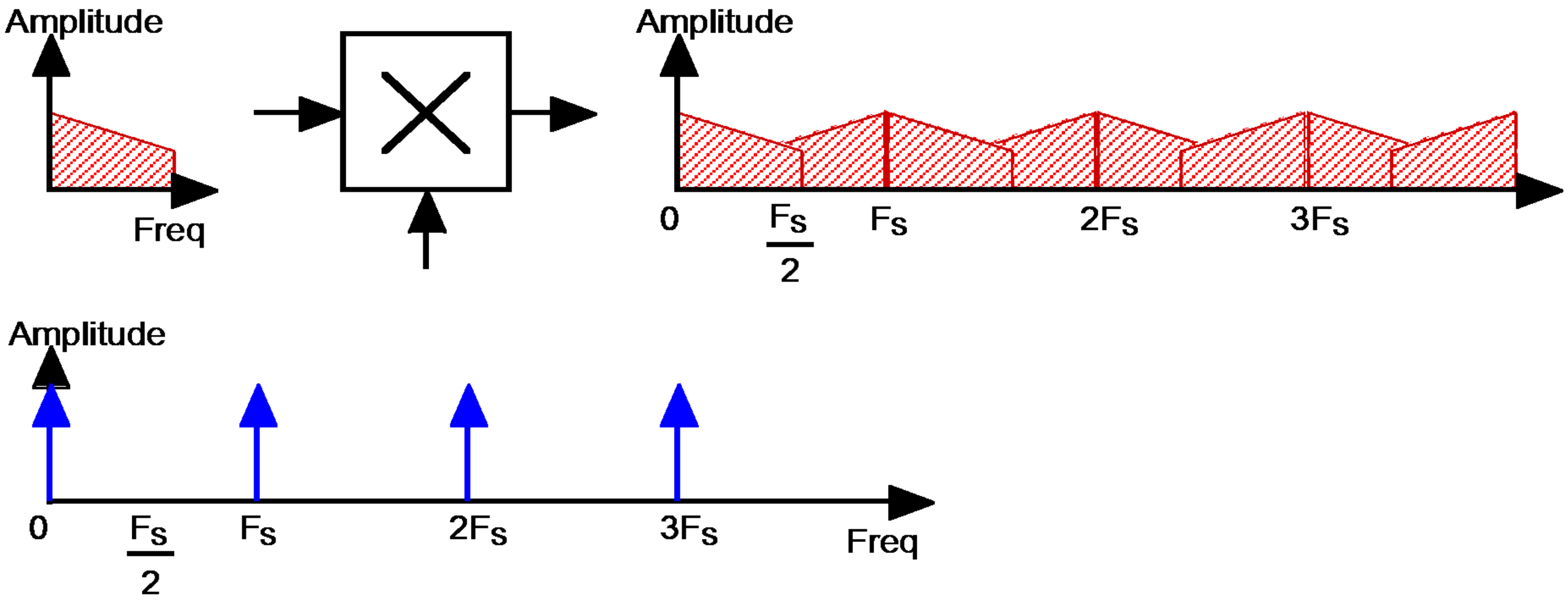


# Sampling In The Frequency Domain

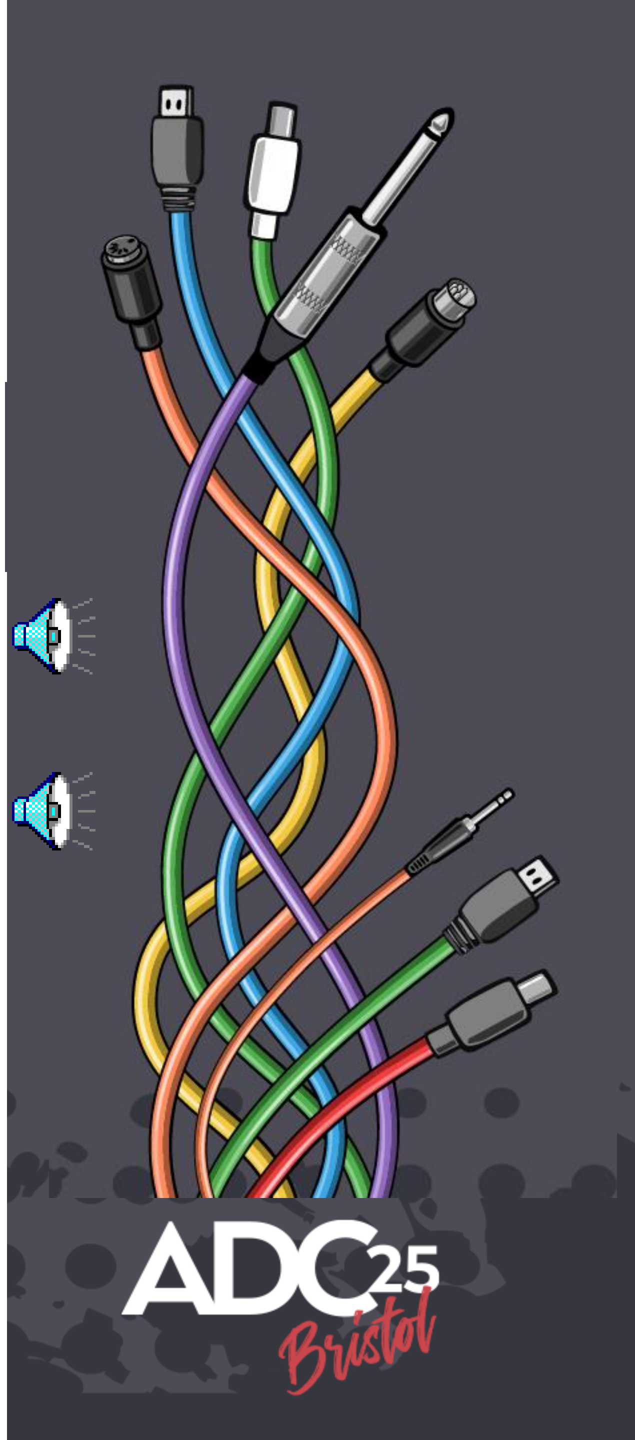



- Must have at least 2xBandwidth.
  - Less than this means the bands overlap
  - And it is impossible to recover the original waveform
  - This is called aliasing.

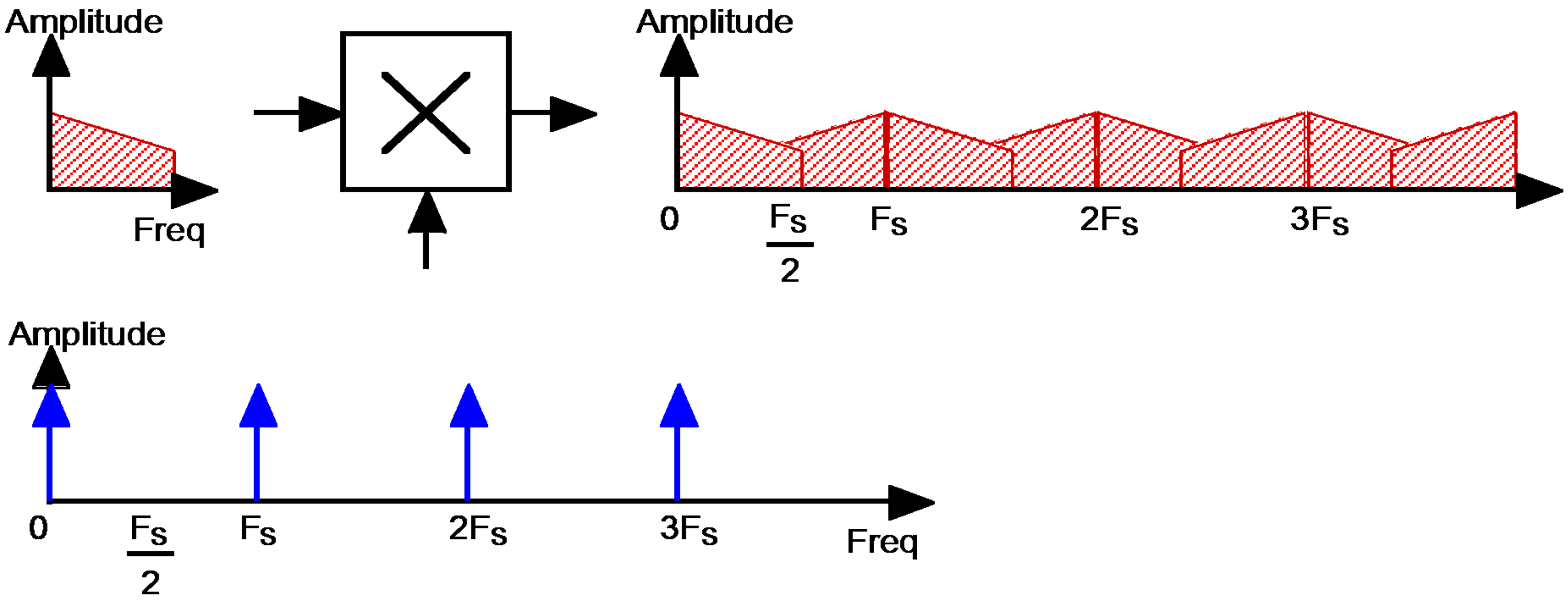
# Sampling In The Frequency Domain




- Must have at least 2xBandwidth.
  - Less than this means the bands overlap
  - And it is impossible to recover the original waveform
  - This is called aliasing.

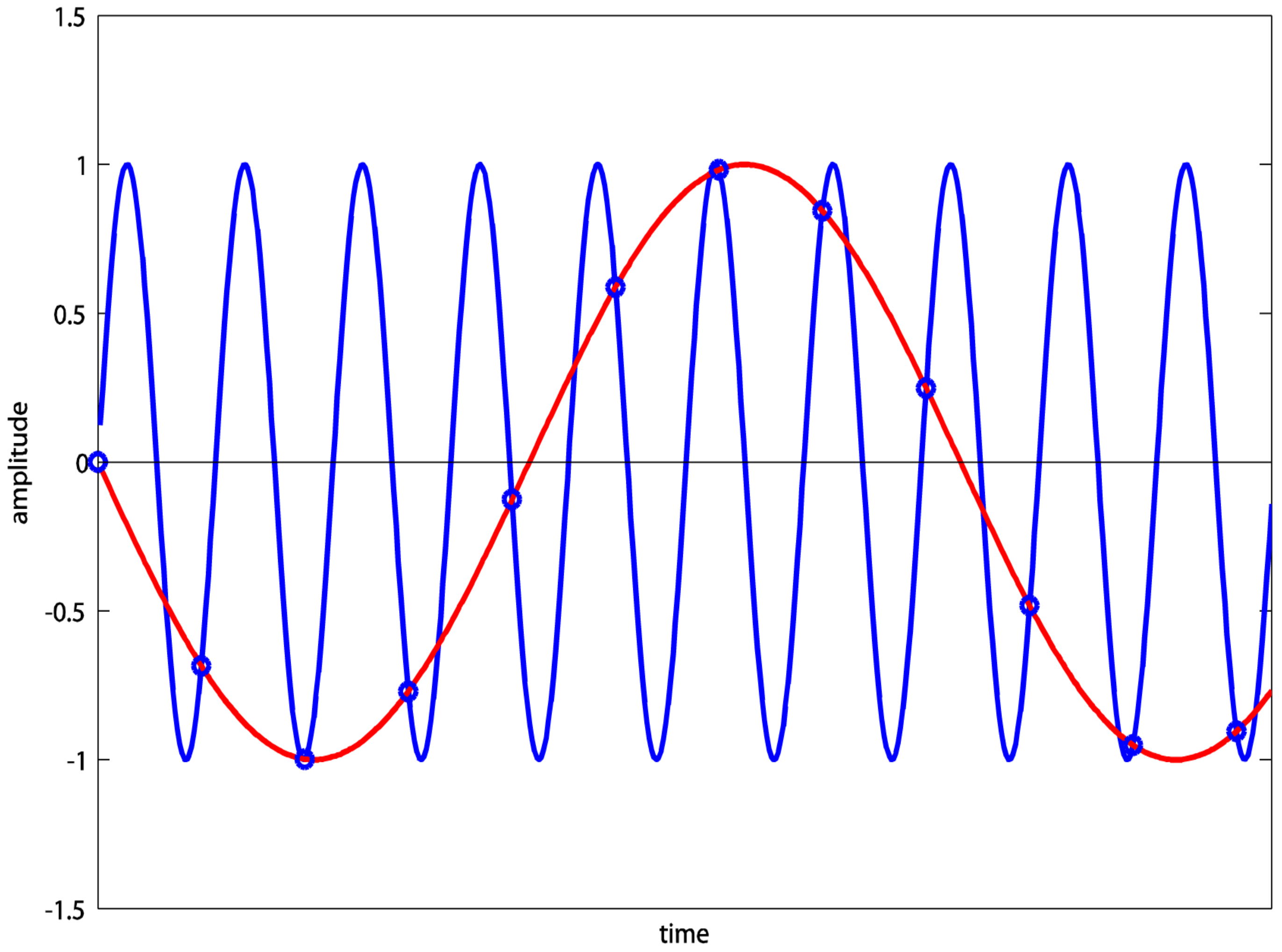


# Sampling In The Frequency Domain

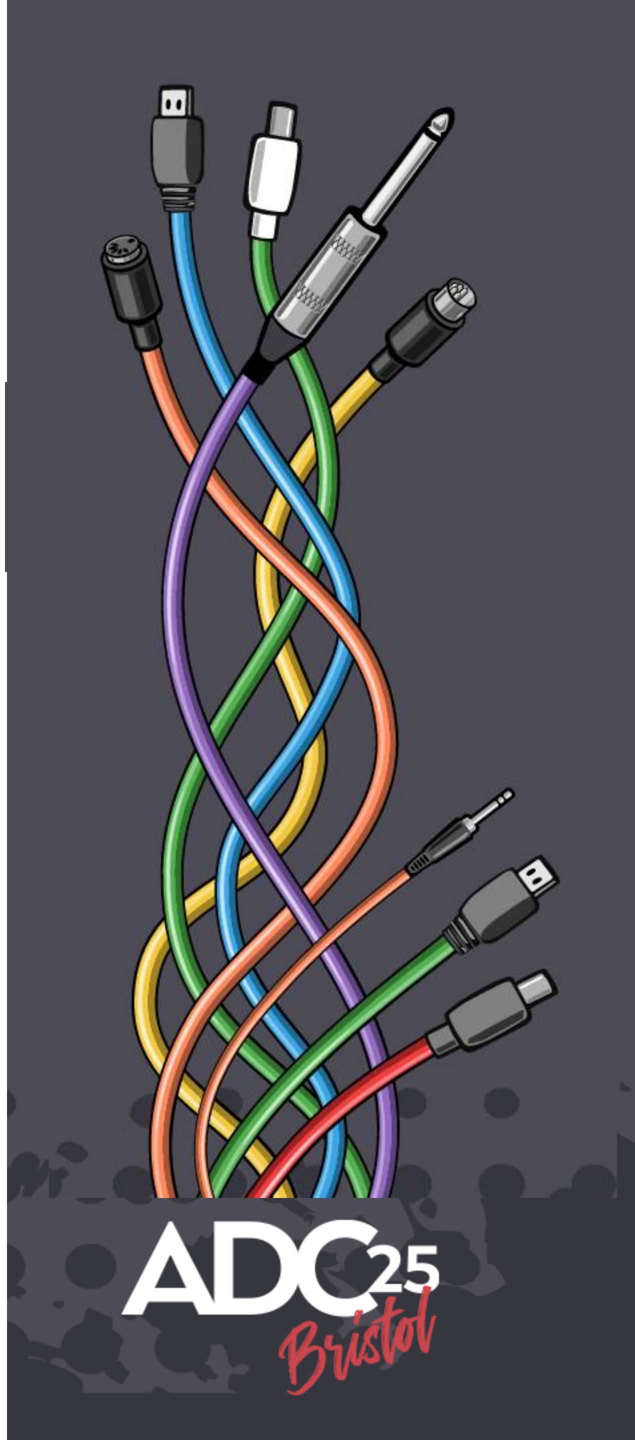



- Must have at least 2xBandwidth.
  - Less than this means the bands overlap
  - And it is impossible to recover the original waveform
  - This is called aliasing.

# Sampling: Time Domain Aliasing




- More than one possible sine wave fits the samples
  – There are an infinite number that do!



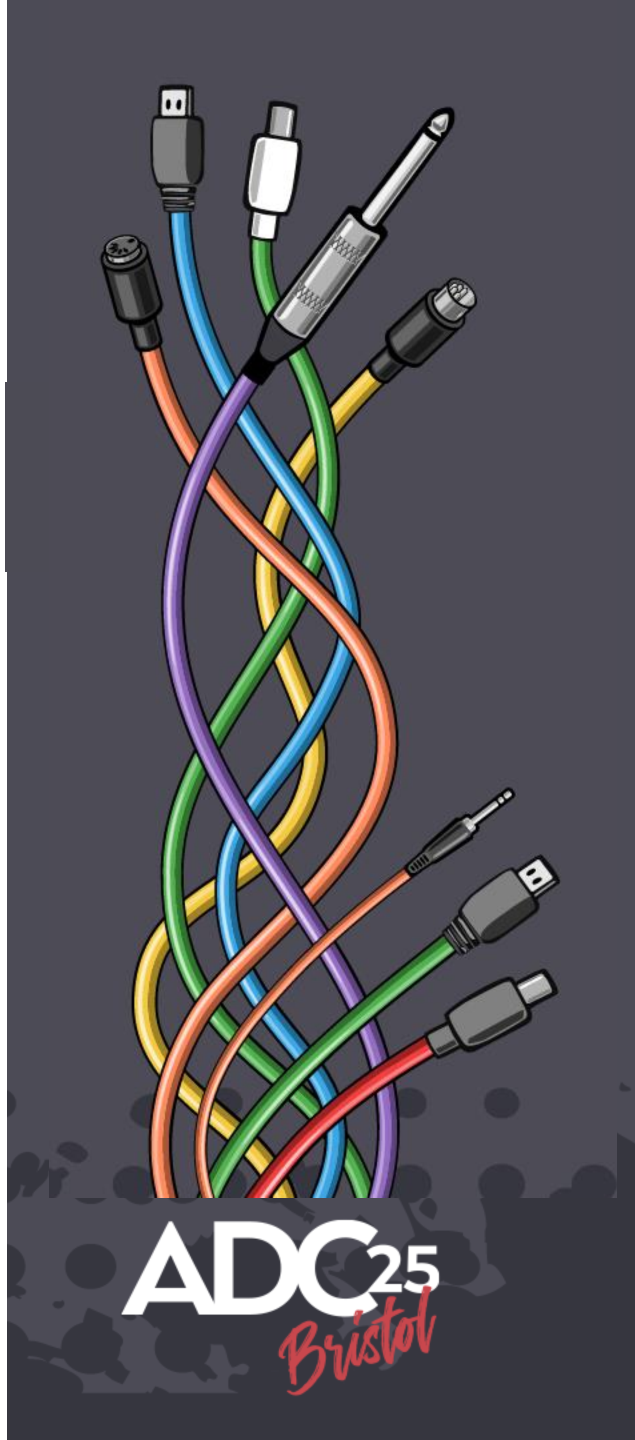
# Sampling: Frequency Domain Aliasing

- Like having a "Through the Looking Glass" mirror.
  - At f=Fs/2,
  - And at f=0
- A sine wave walks into the mirror…

- And one walks out!
- You see an infinite number of reflected baseband spectra in two parallel mirrors

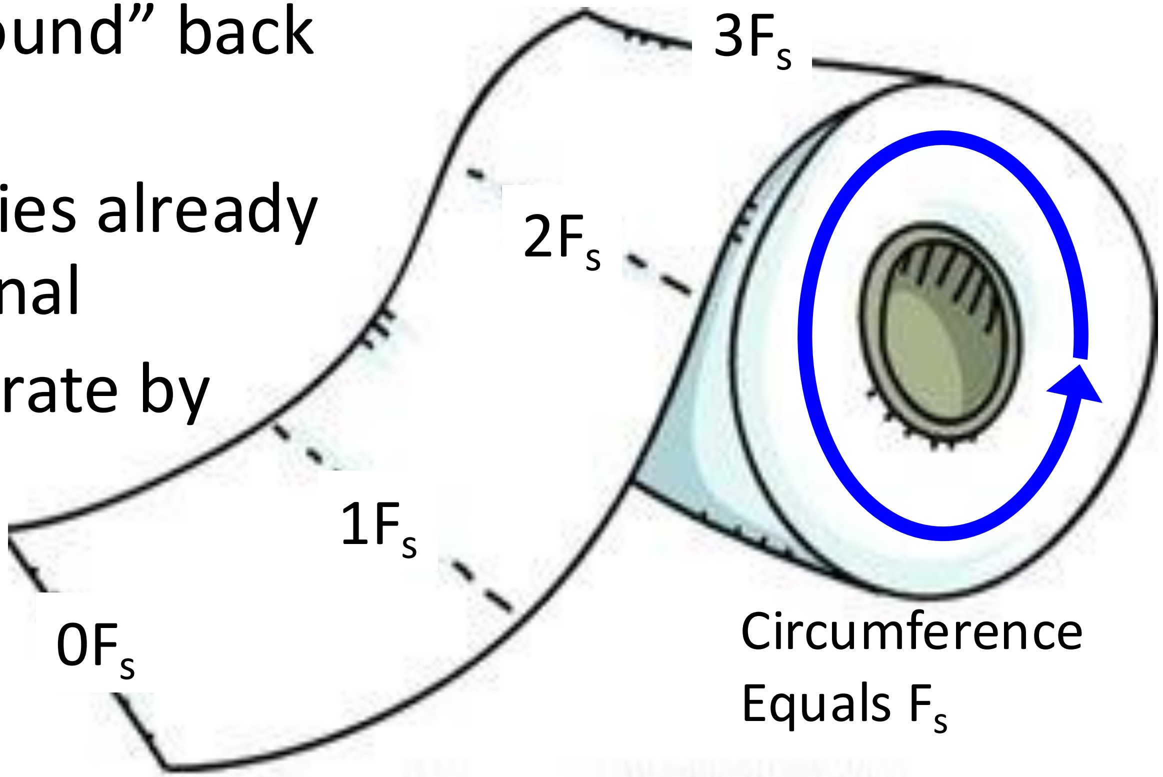
# The Result of Sampling

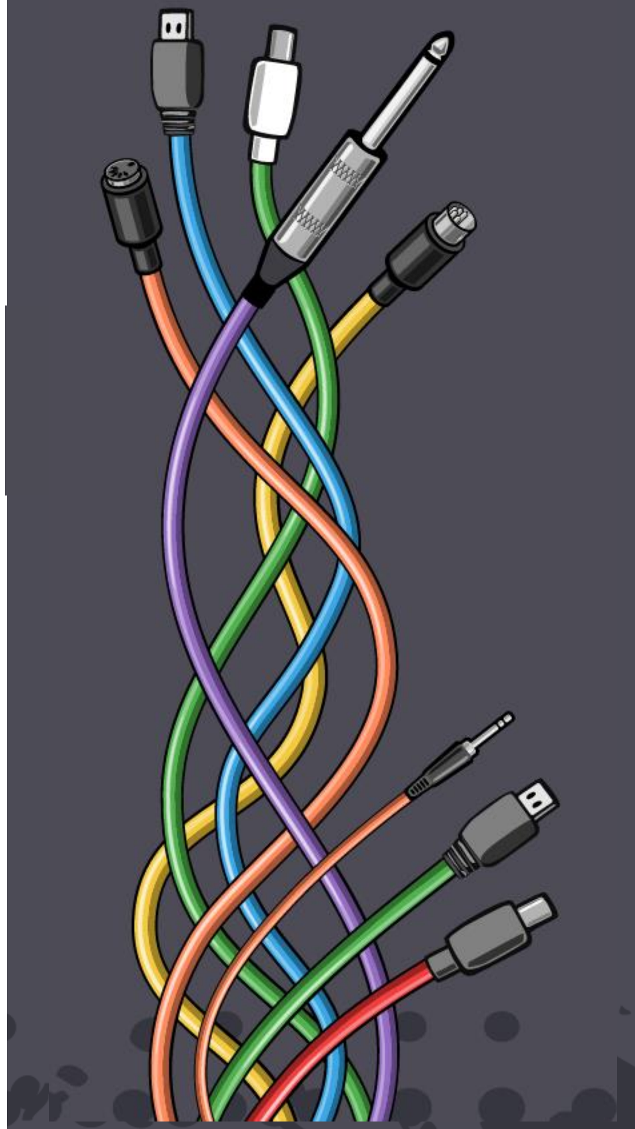
- After sampling, the infinite frequency range of the real world.
- The infinite frequency of:
  - minus infinity
  - to plus infinity
- Collapses onto sampled frequency of:
  - minus Fs/2
  - to plus Fs/2
- Or, more conventionally, 0 Hz to Fs Hz
- Which is why the samples are not what they seem!



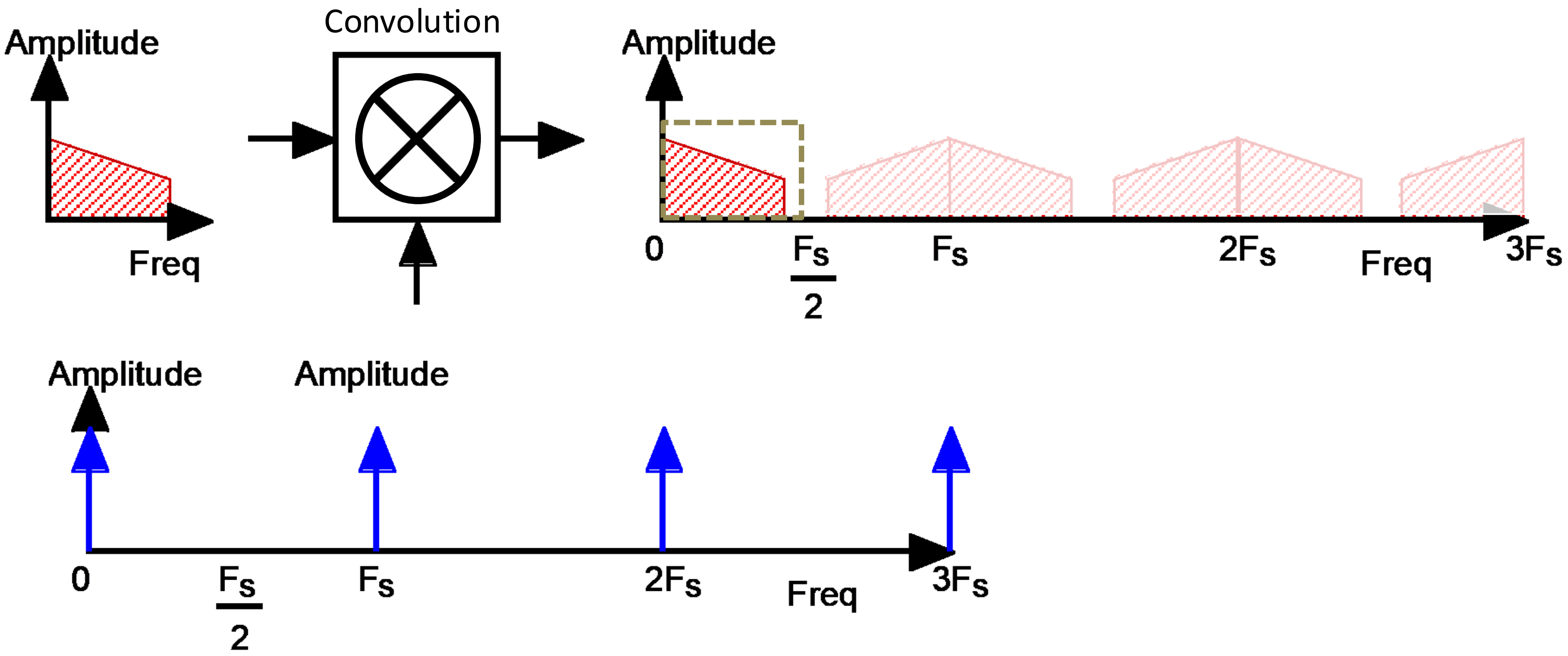
# The Result of Sampling

- **0 to Fs is all you have!**
  - But you can only use half of it
  - Any frequencies that are outside that range
  - Are “wrapped around” back into it!
  - True for frequencies already present in the signal
  - Or ones you generate by distortion!
  - We need to unwrap the toilet roll!

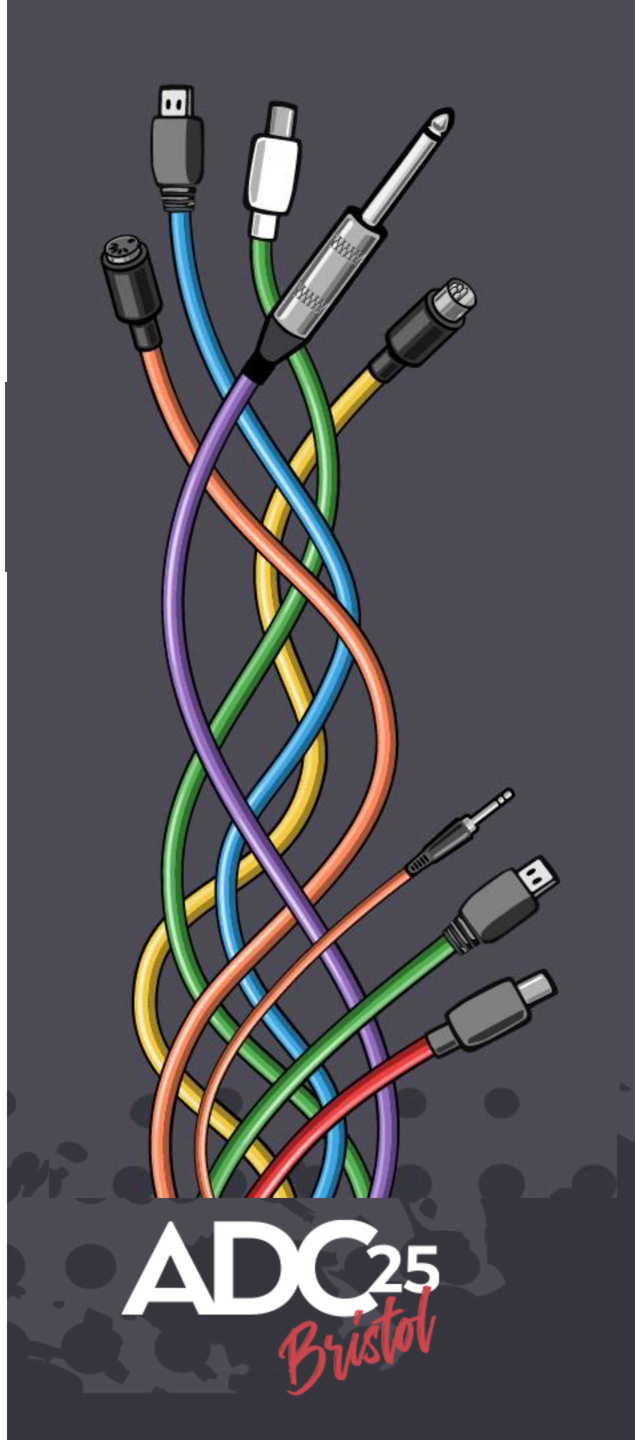


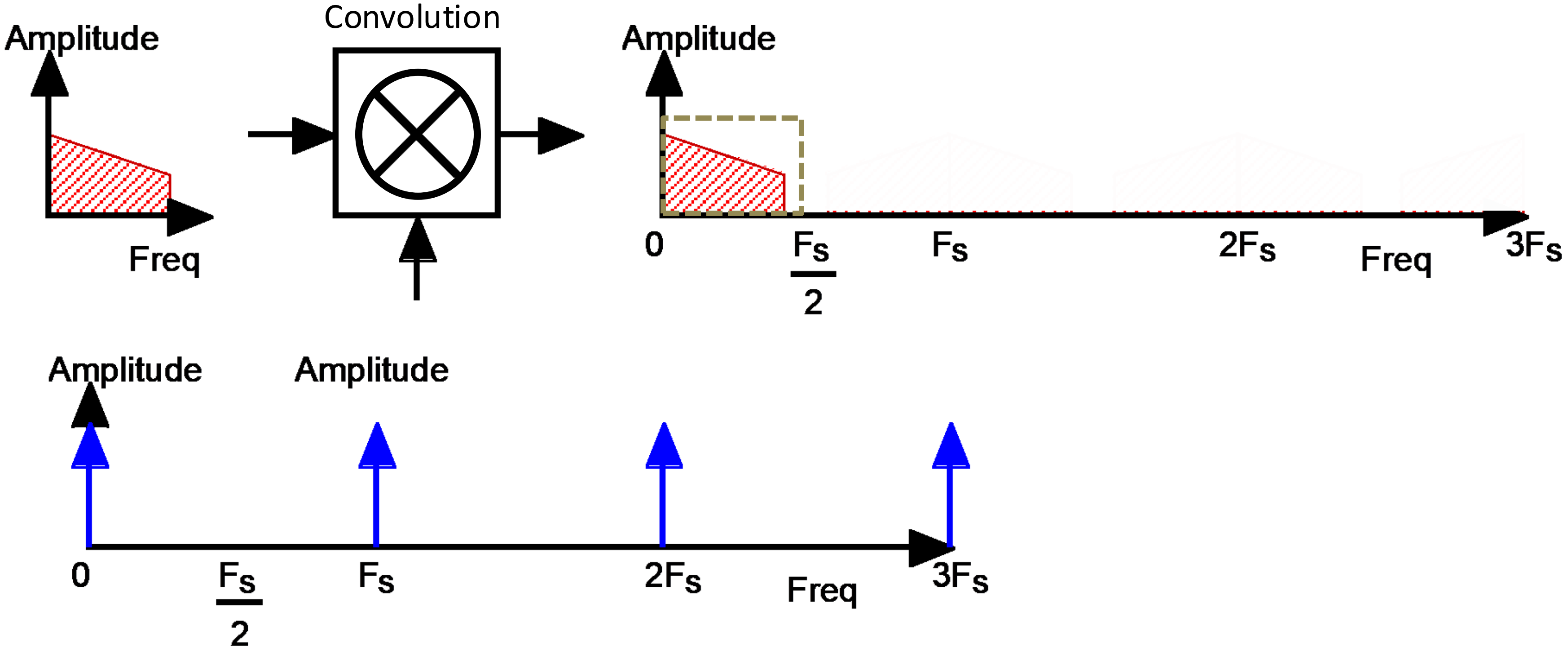
# Getting The Waveform Back: Reconstruction



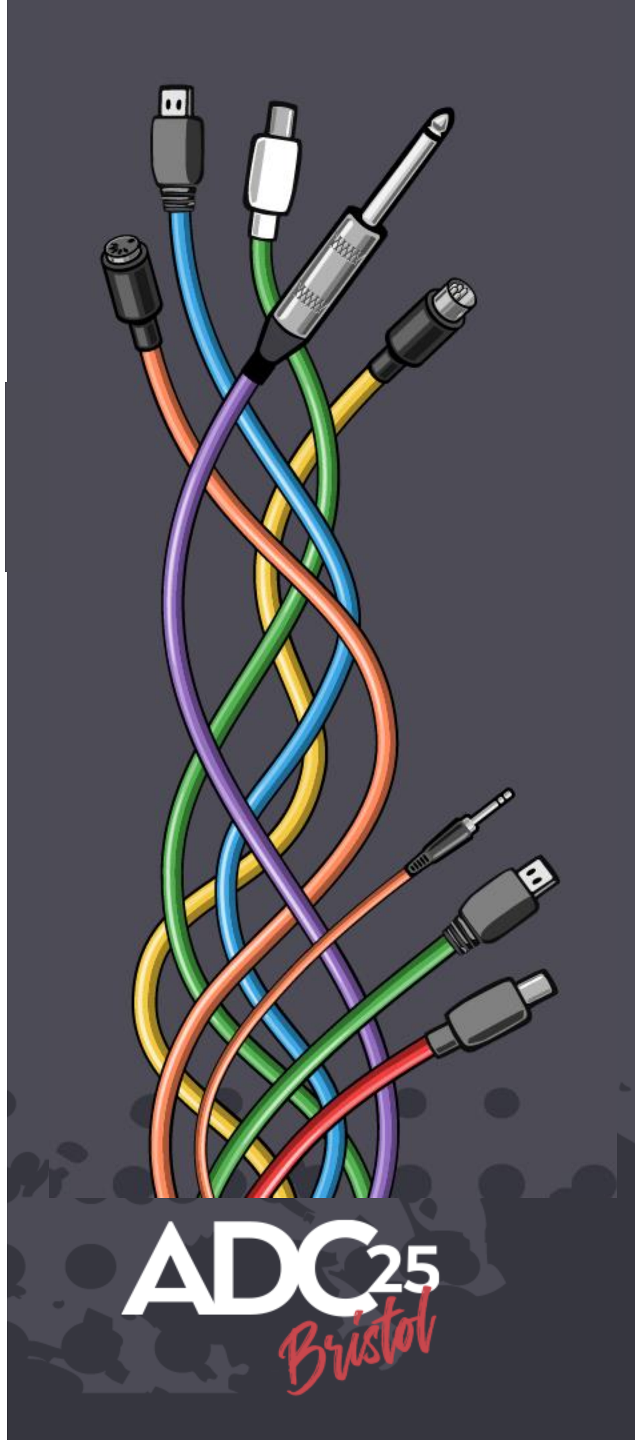

- Removing sampling aliases
  - In the frequency domain
  - The baseband can be separated from the aliases
  - By filtering with a perfect low-pass reconstruction filter



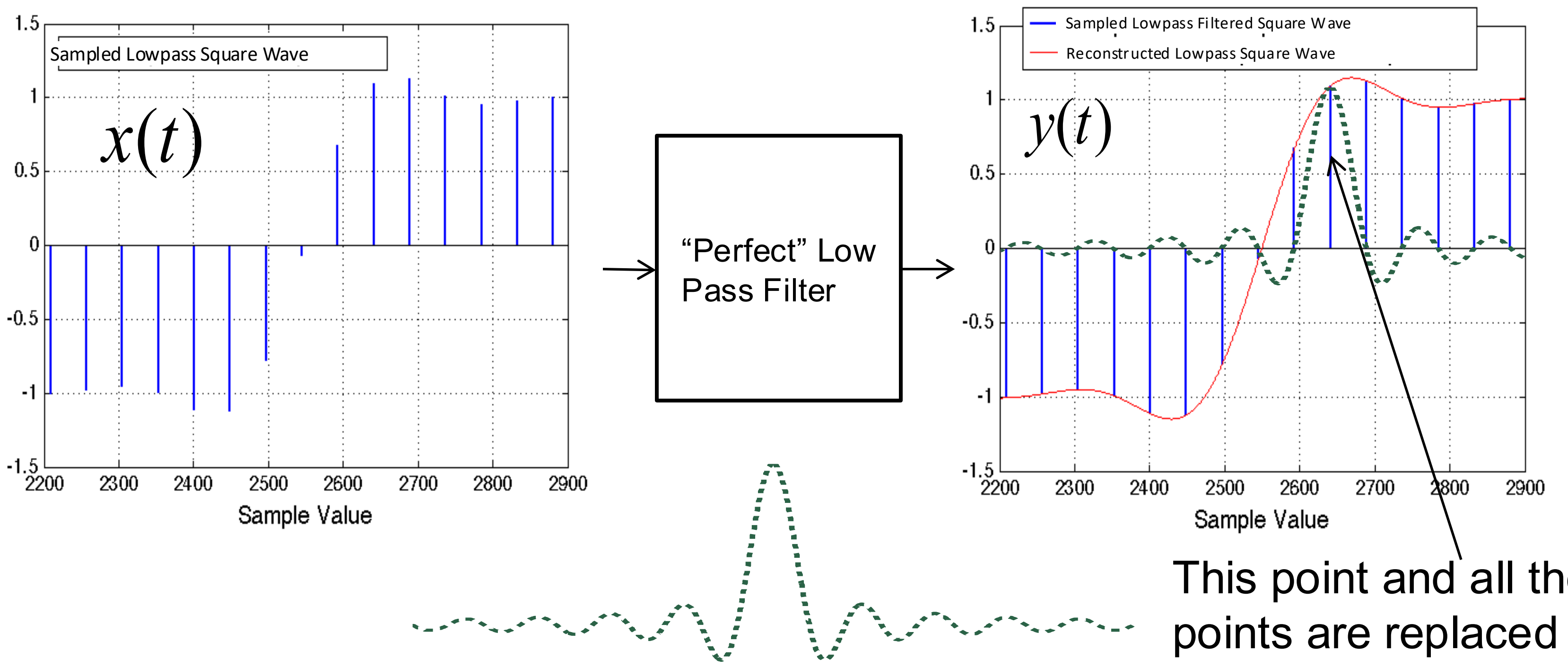
# Getting The Waveform Back: Reconstruction




- Leaving the baseband
- Which is the original signal before sampling
- But what does this look like in the time domain?



# Getting The Waveform Back: Reconstruction

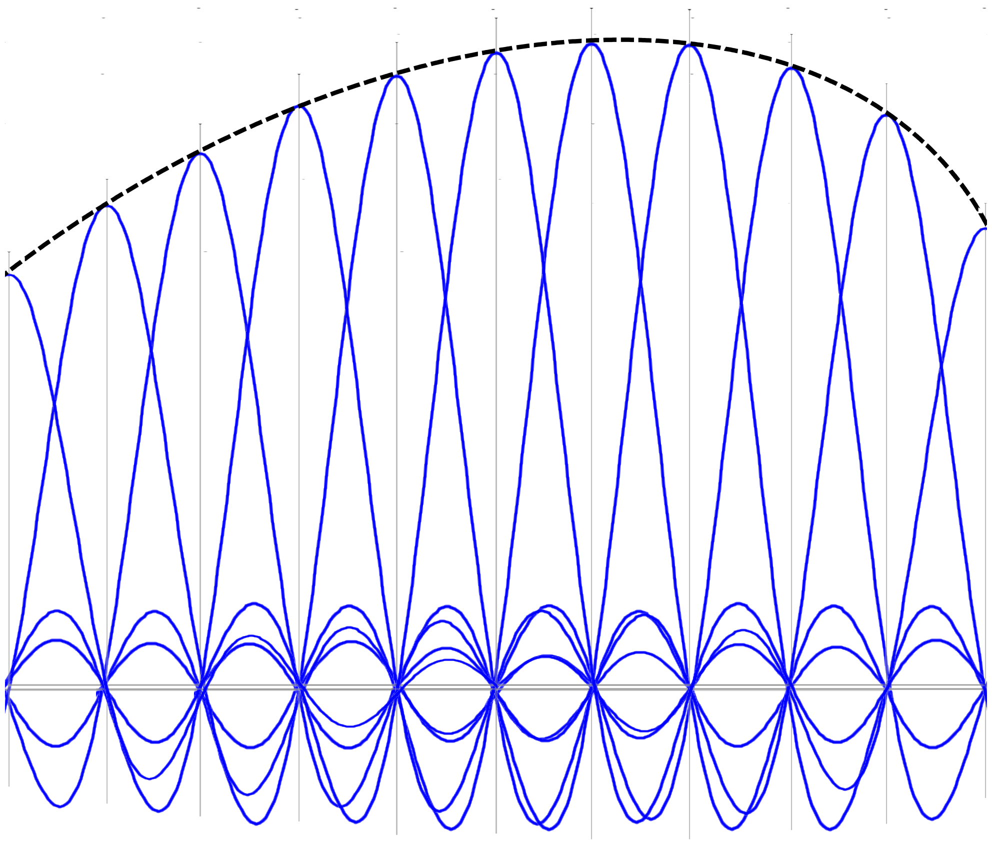



"*impulse response*" $h(\tau)$ = Sinc($\tau$)

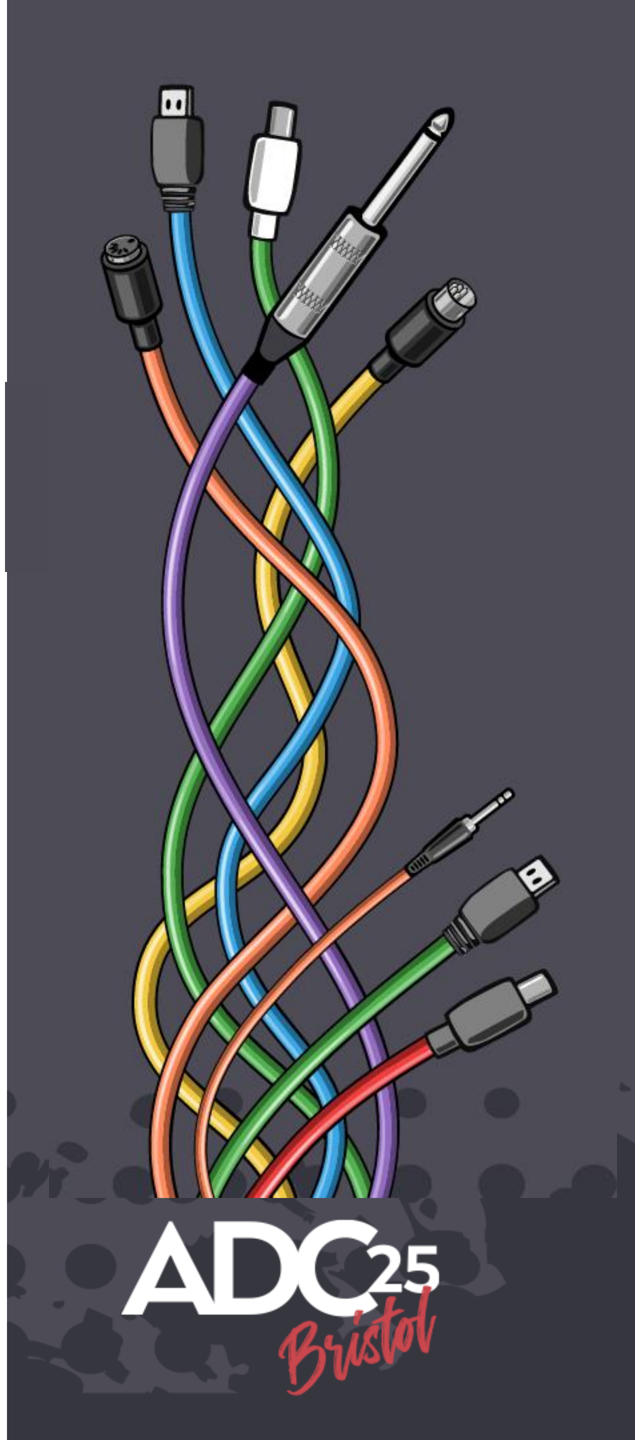
This point and all the other points are replaced by the impulse response.

- The reconstruction filter replaces the samples with the impulse response to re-form the original input signal.
- Regular sampling has a simple, linear, reconstruction process

# Getting The Waveform Back: Reconstruction

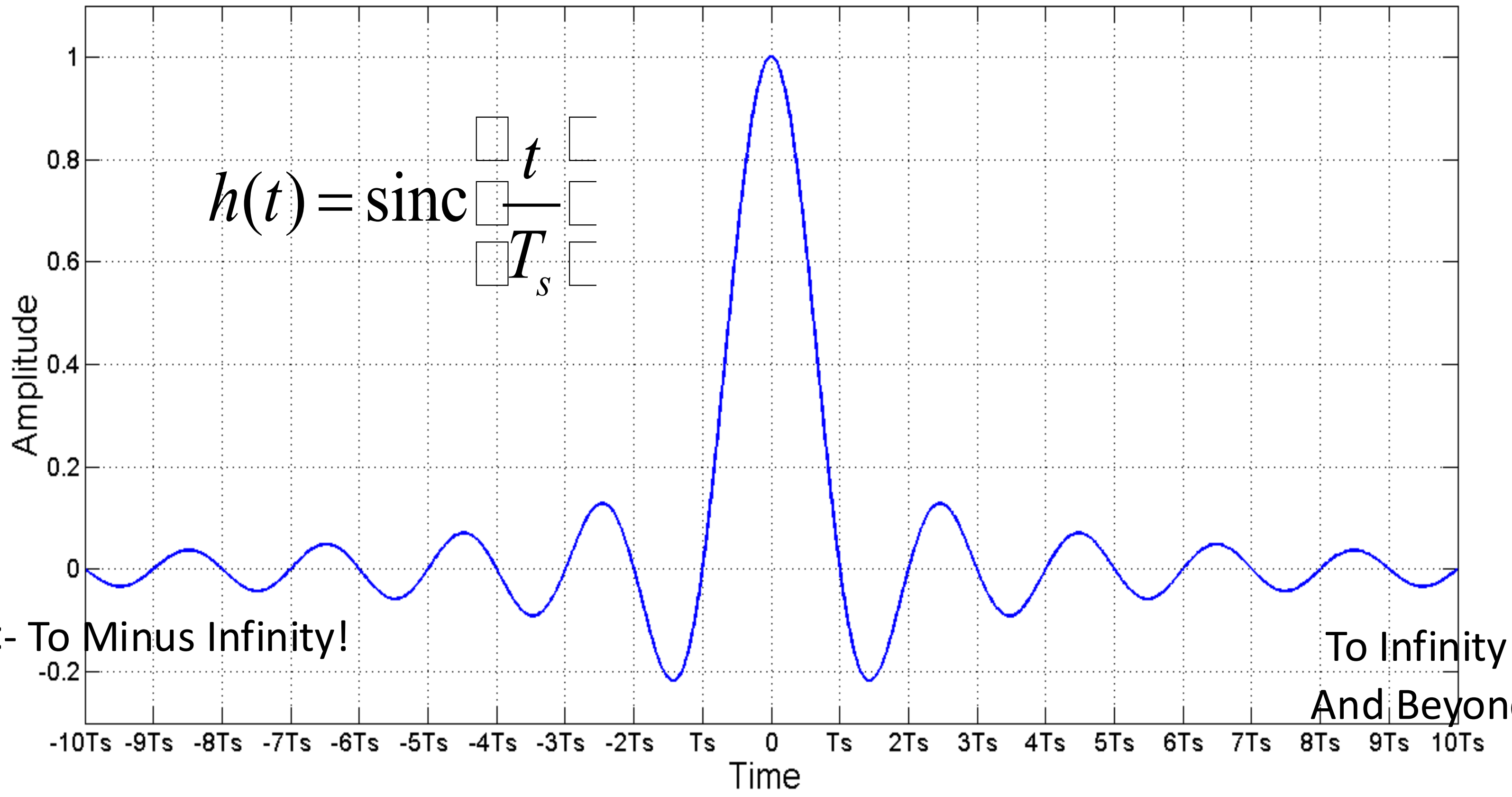


- Because the reconstruction filter has zeros at every other sampling point
  - Besides the sample point



# Reconstruction Filter Impulse Response

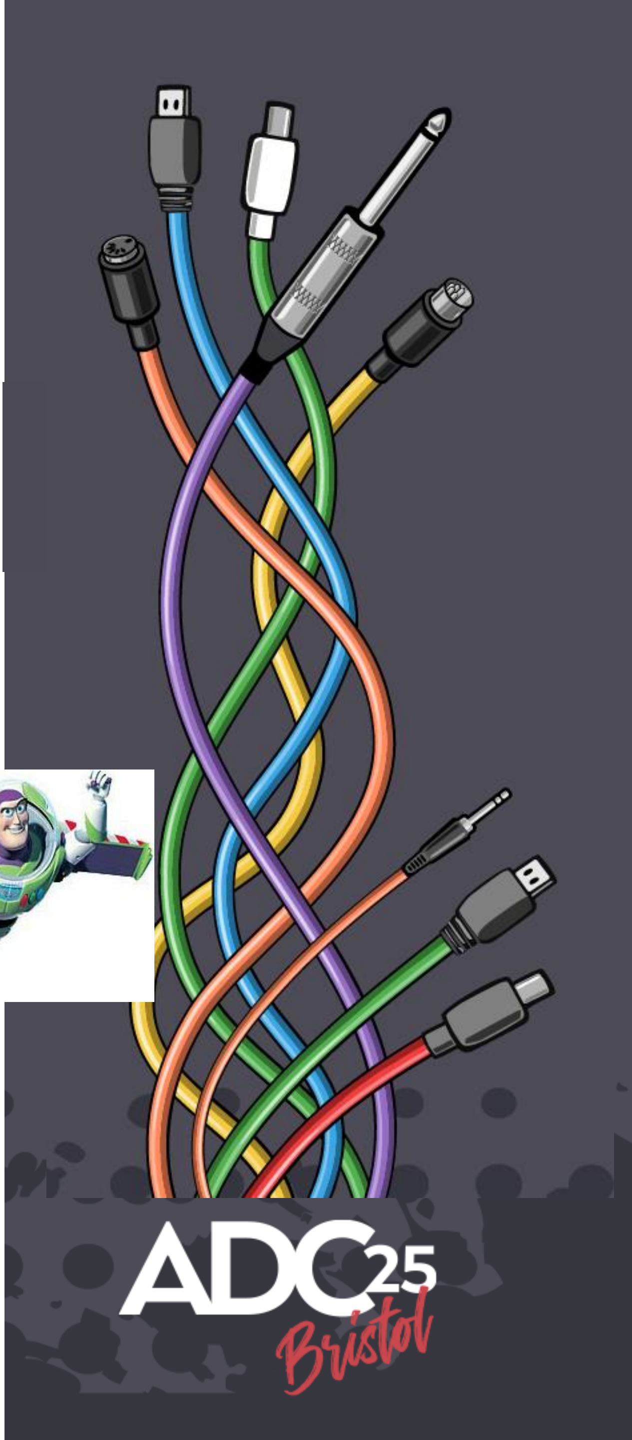
$$h(t) = \mathrm{sinc}\left(\frac{t}{T_s}\right)$$



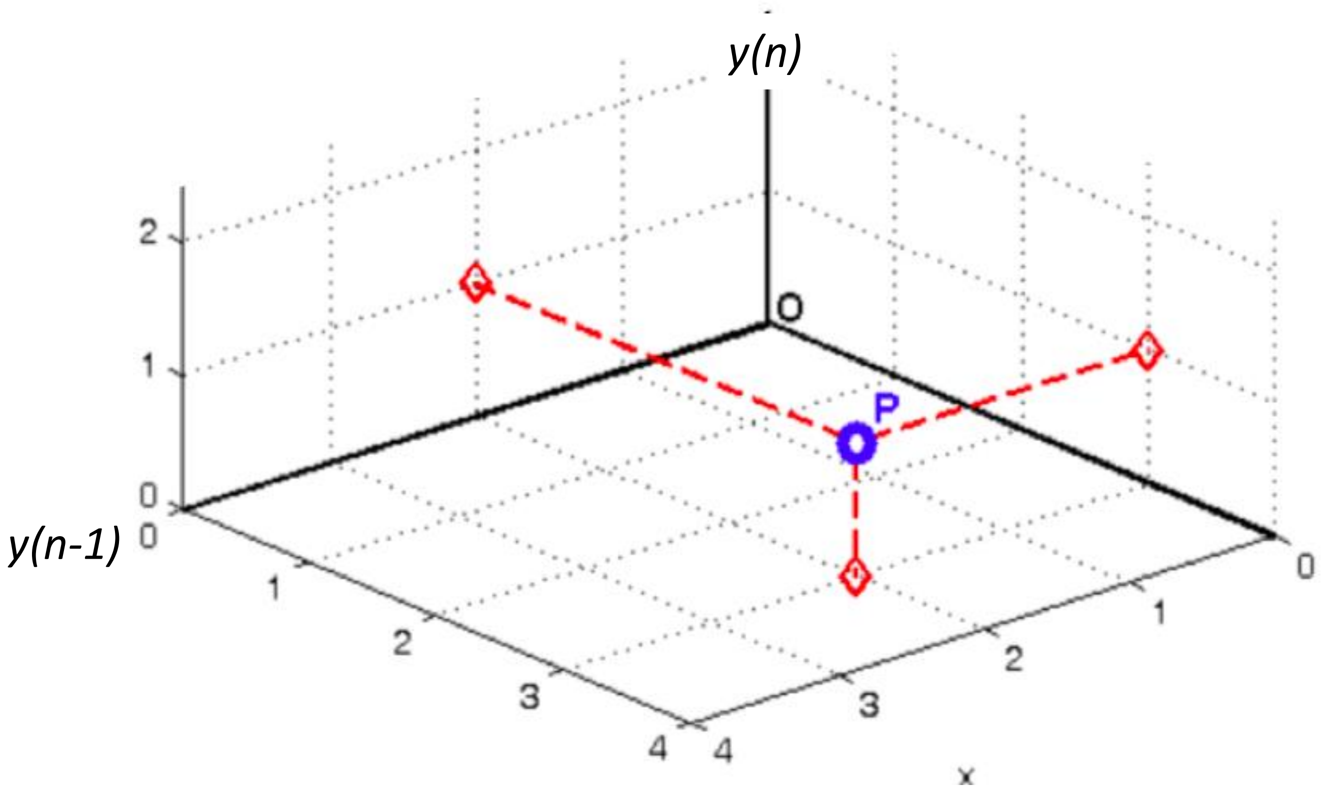

<- To Minus Infinity!

To Infinity! ->
And Beyond!

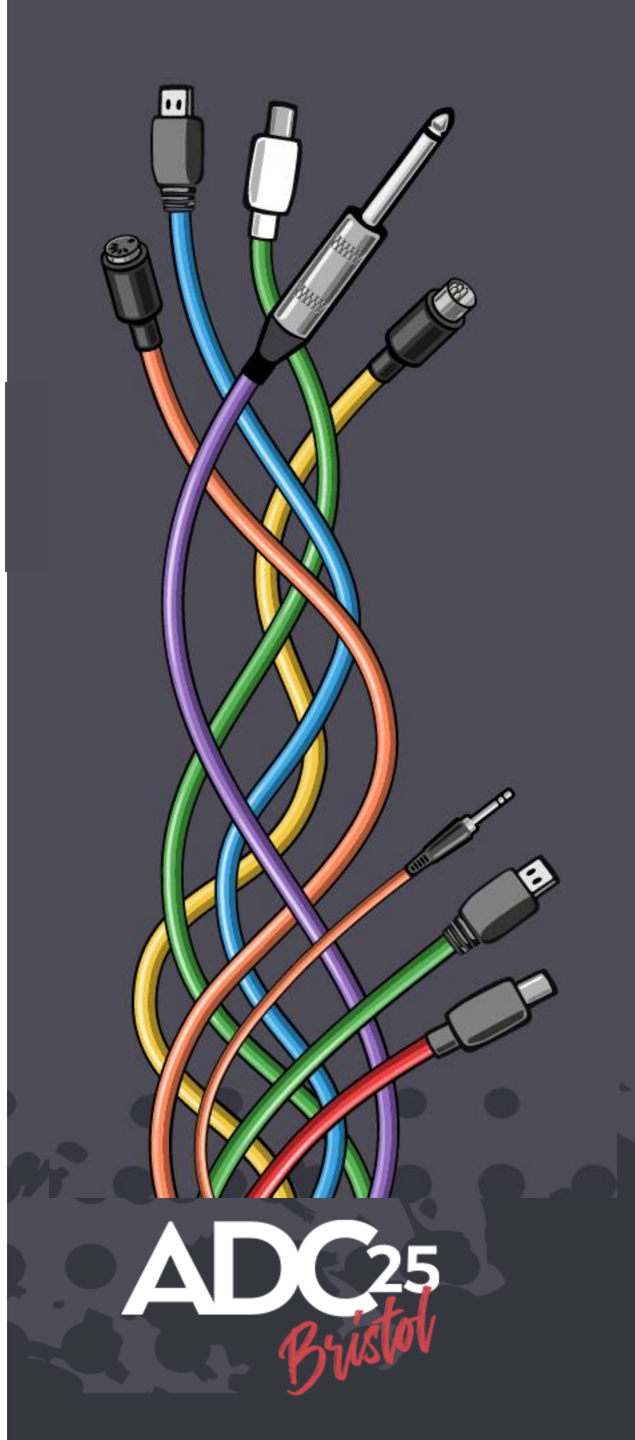
- Crosses zero at all sample times ≠ 0
- It is orthogonal to itself for samples $T_s$ seconds apart
- This means the sample values can be seen as coordinates!
- However, the impulse response itself is infinitely long…



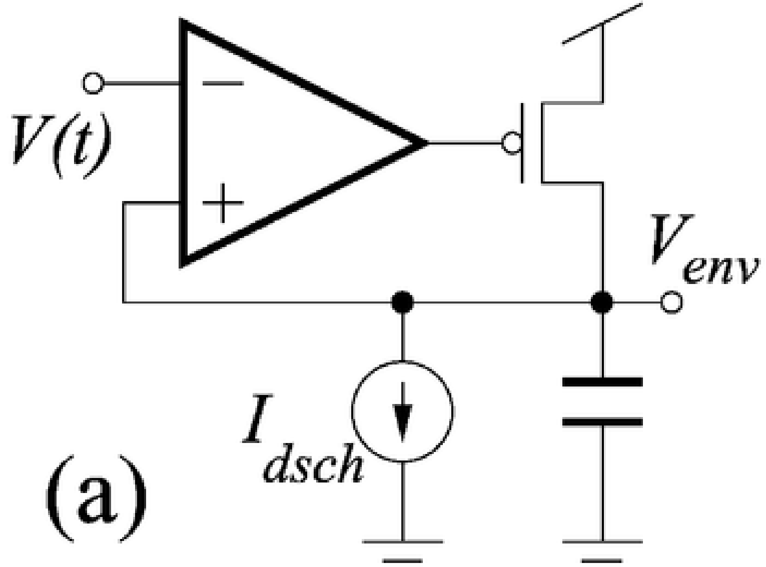
# Getting The Waveform Back: Reconstruction

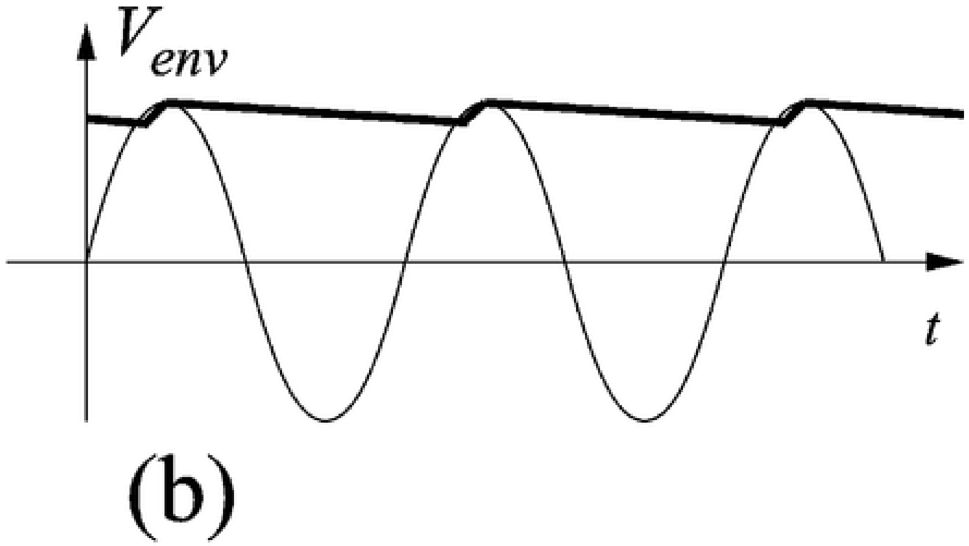



- Samples of the filtered signal are orthogonal
  - Thus, a sequence of samples specifies a unique point
  - One point to rule them all…
  - That contains **ALL** the information about the signal
  - Thus, one can perfectly reconstruct the original signal



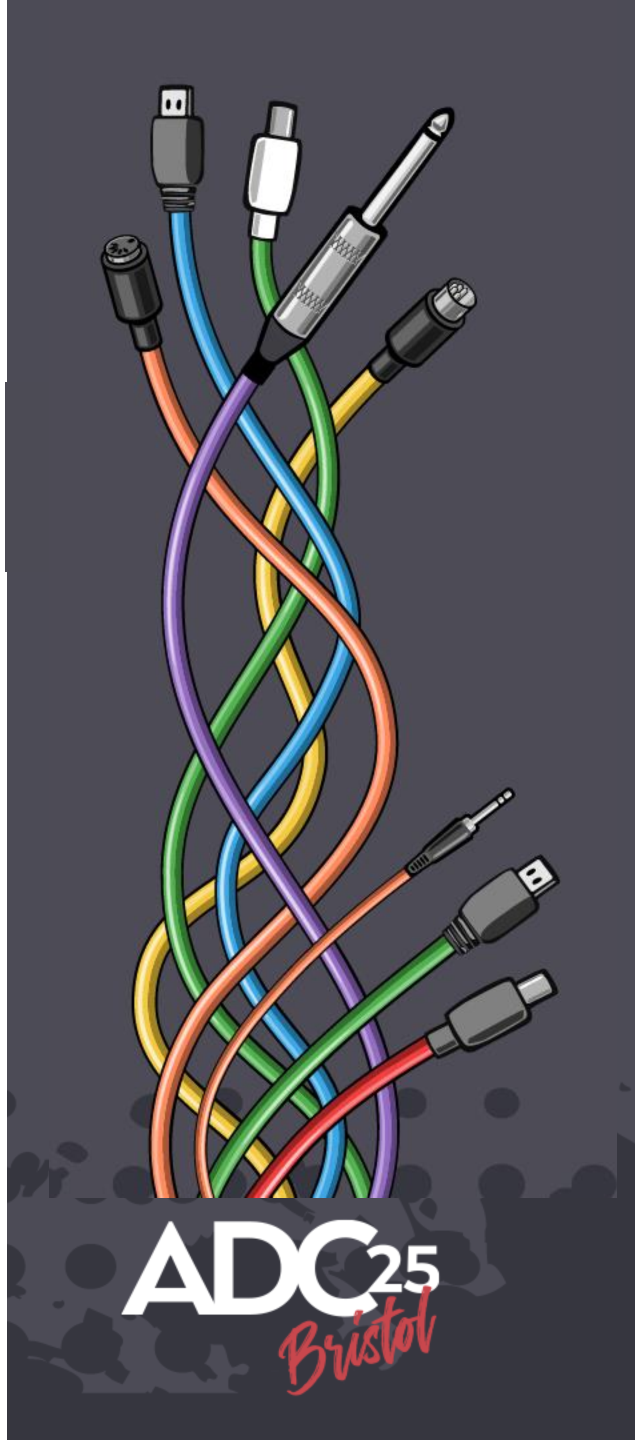
# Analogue vs Software Peak Detector



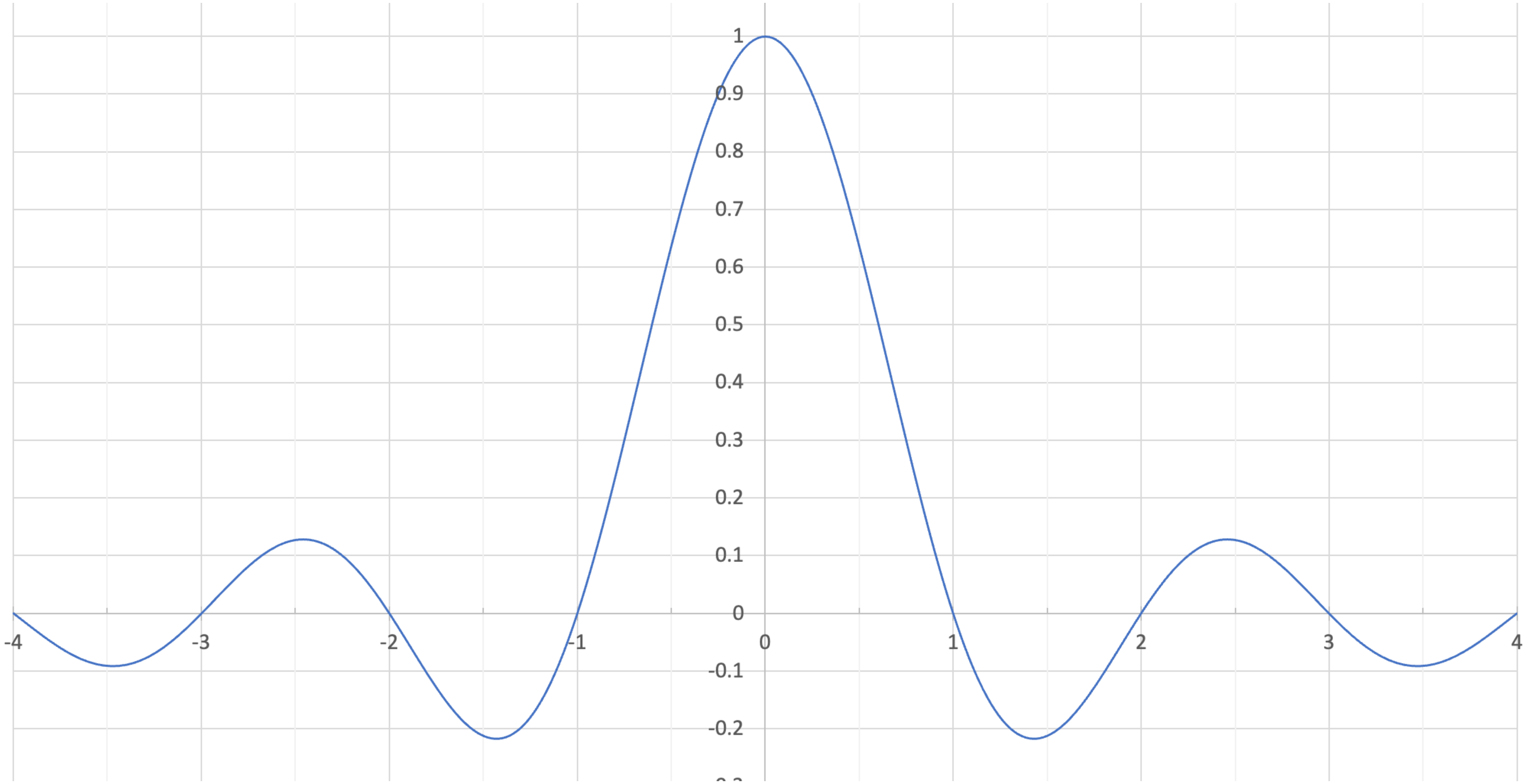




- The analogue peak detector can see the whole waveform
- The software one can only see sample values

```
Input: signal
Output: index
peakIndex := null;
peakValue := null;
for index, value in signal do
    if peakValue == null or value > peakValue the
        peakIndex := index;
        peakValue := value;
    end
end
return peakIndex;
```

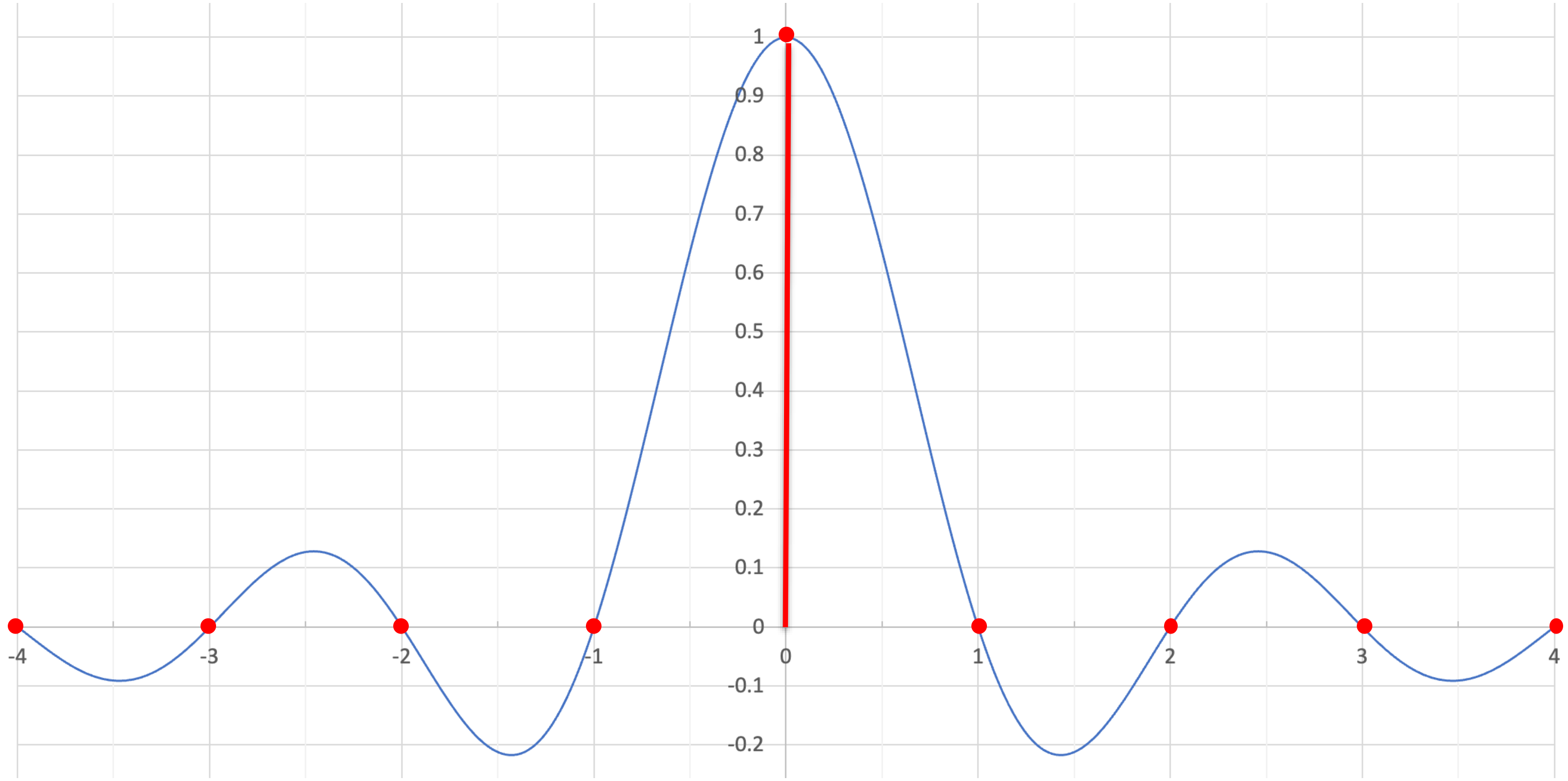


# Band Limited Impulse

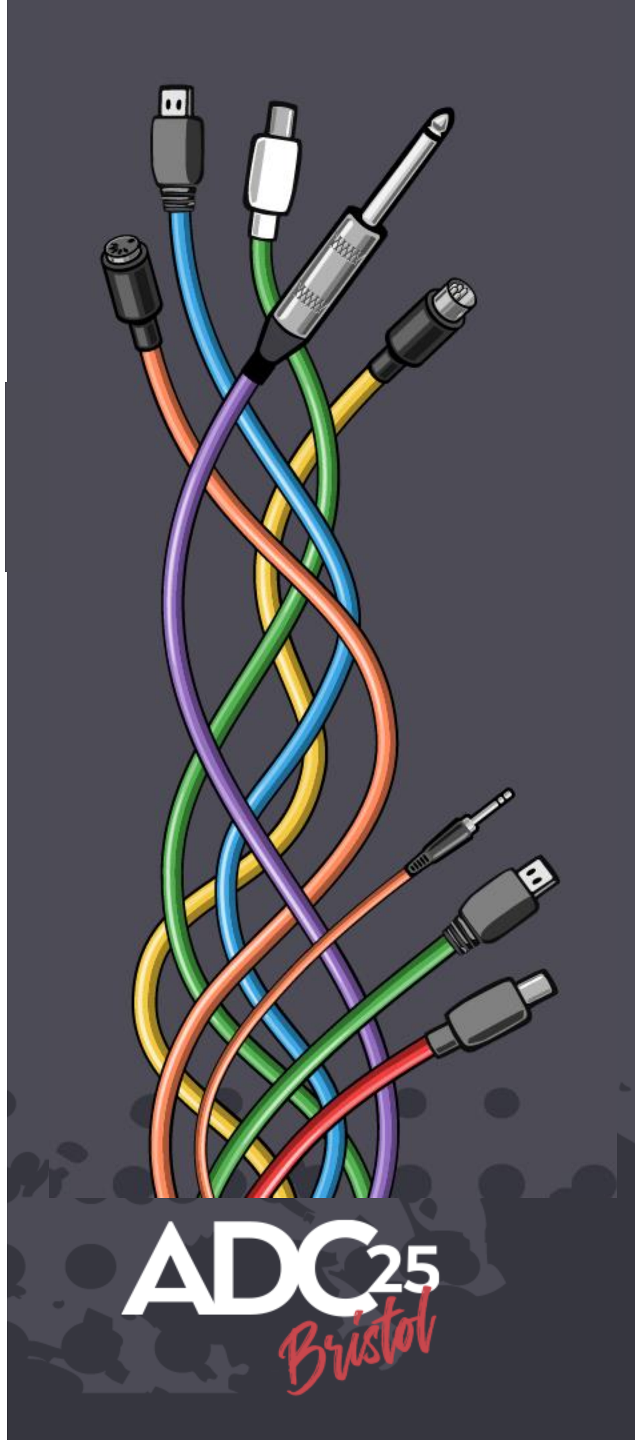



- So how bad can the error be?
- Let’s look at the worst case
  - A band limited impulse

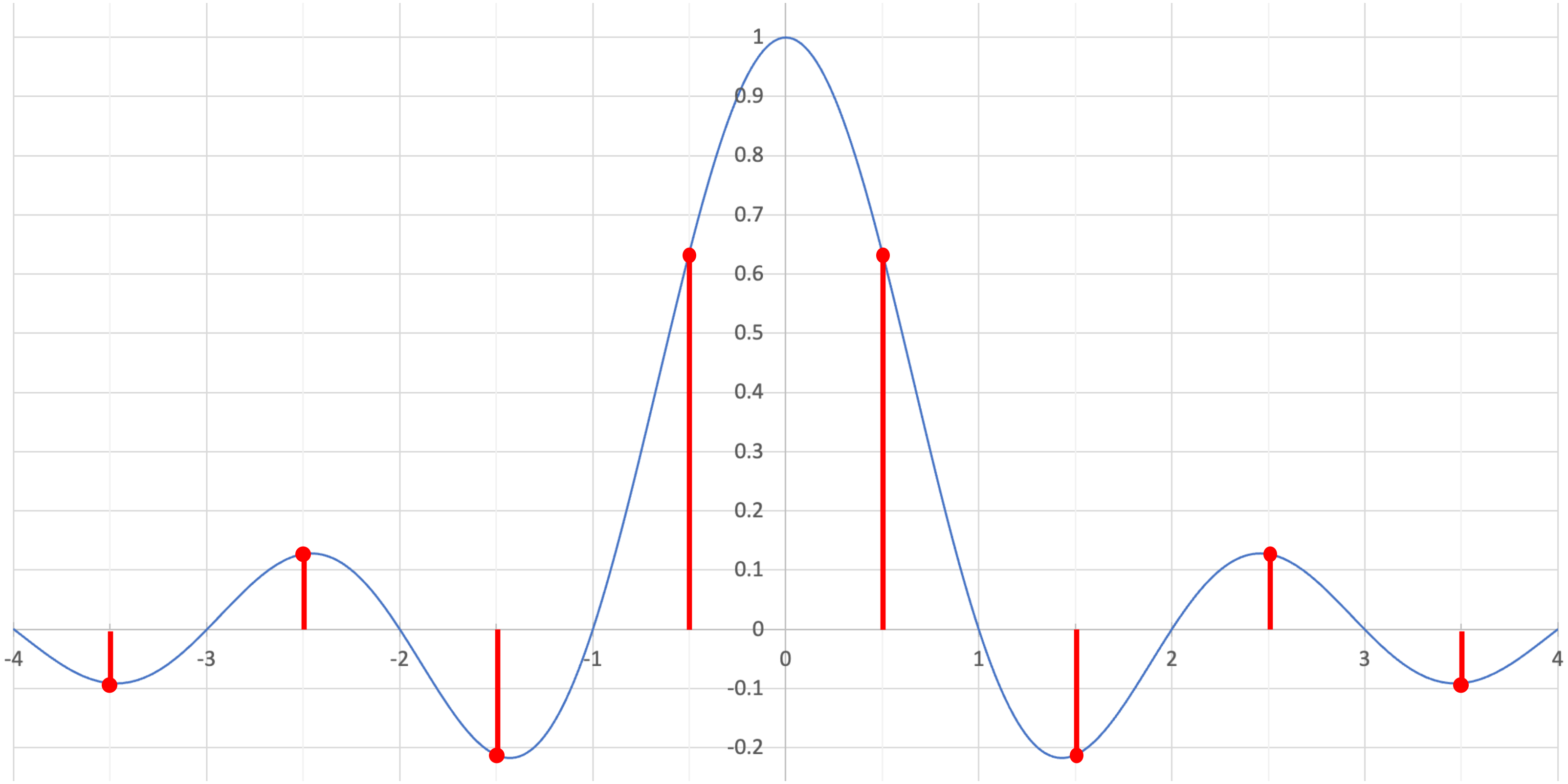
# Digital Impulse Perfect Sample Phase



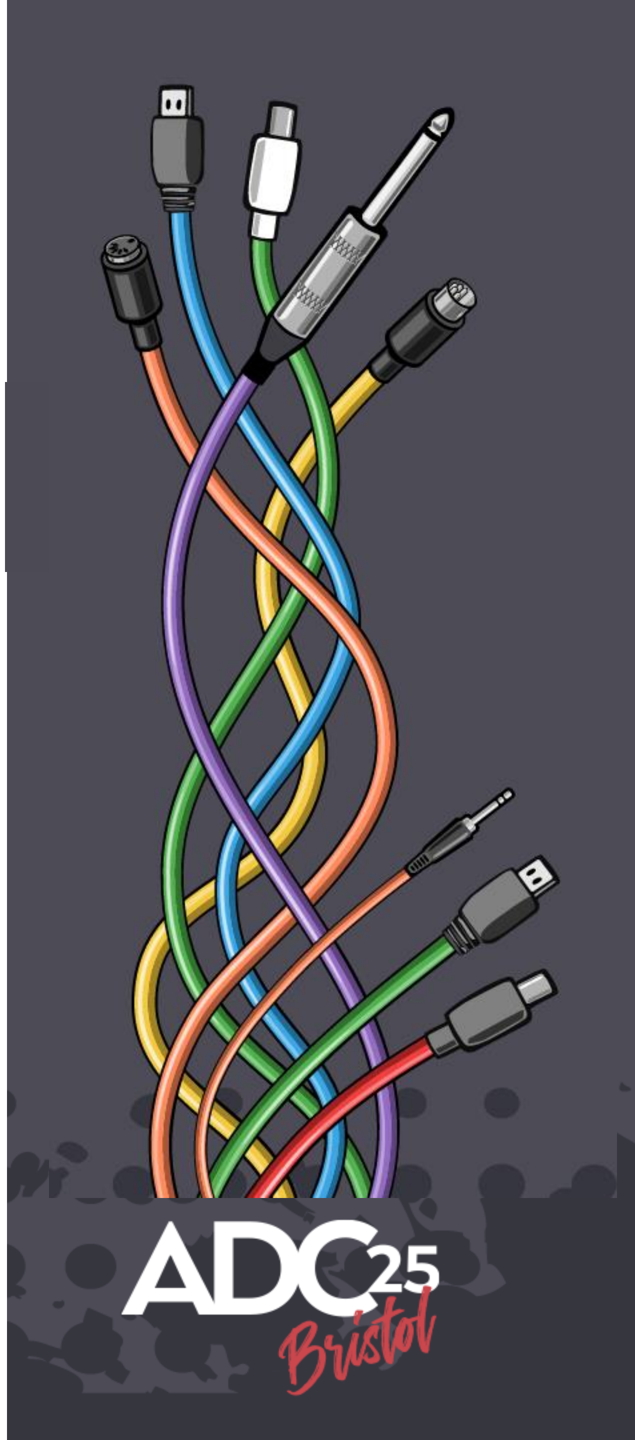

- If the impulse is perfectly sampled in the middle
- There is no error
  - The sample value is the peak of the impulse



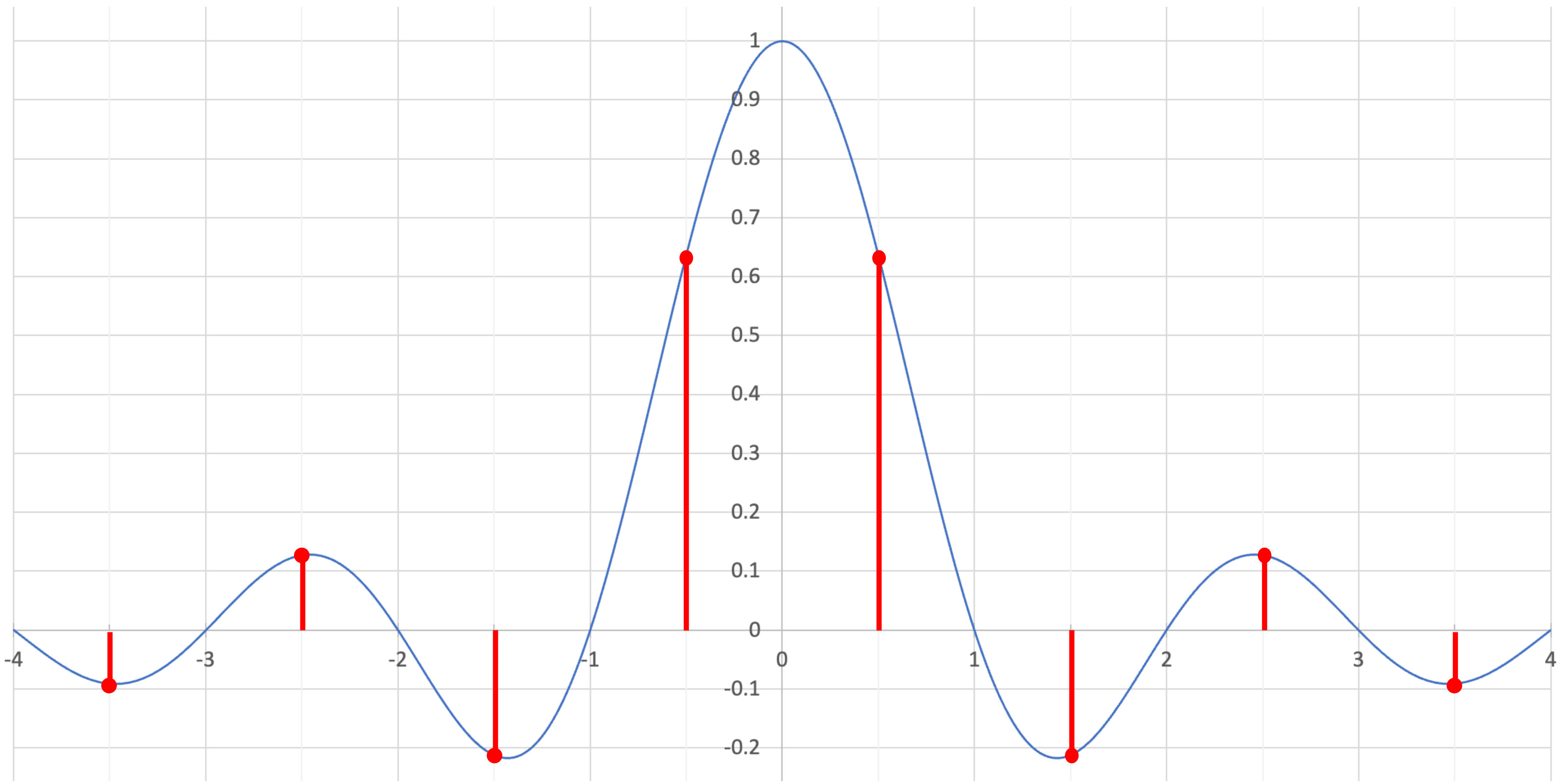
# Digital Impulse Different Sample Phase




- But if the impulse is not sampled in the middle
- There will be an error
  - The sample value is not the peak value of the impulse

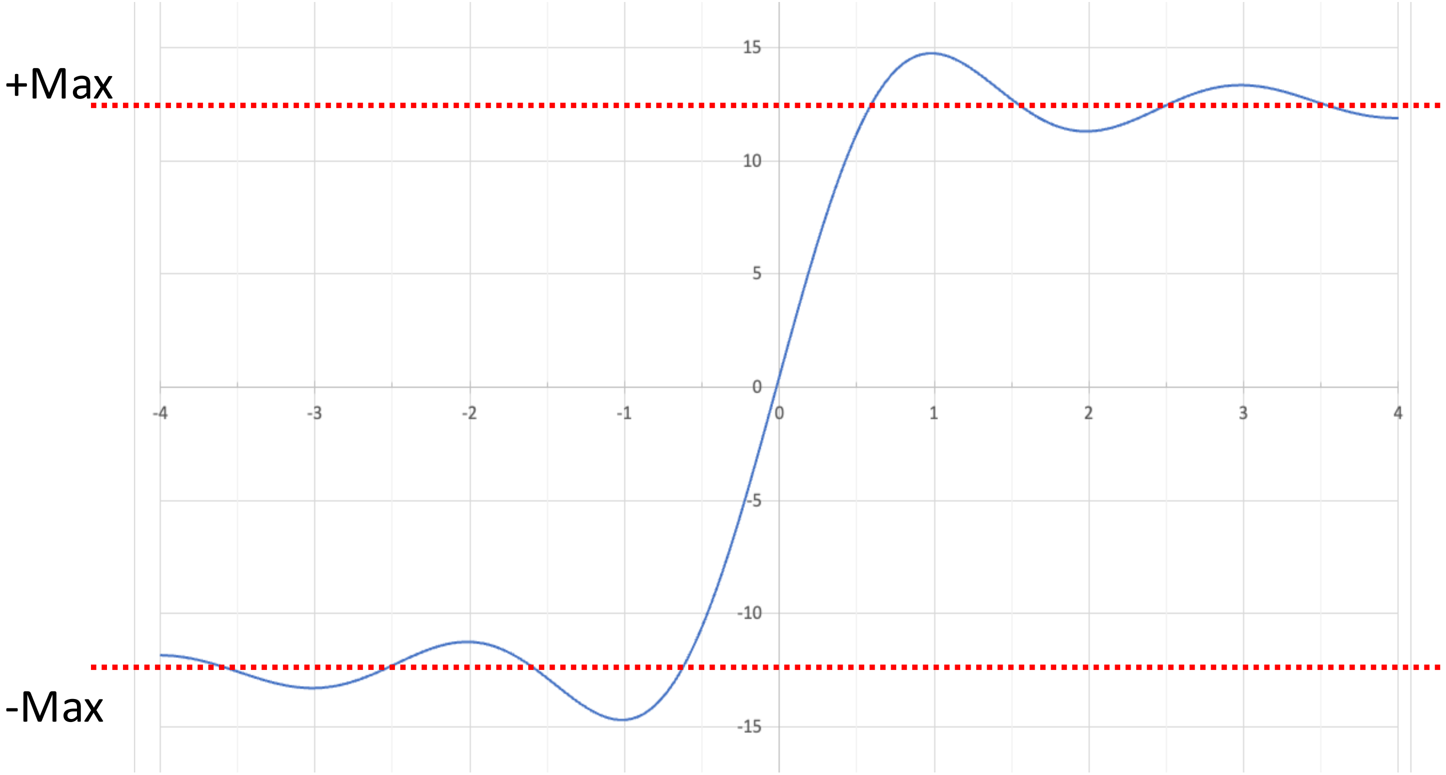


# Worst Digital Peak Detector Error



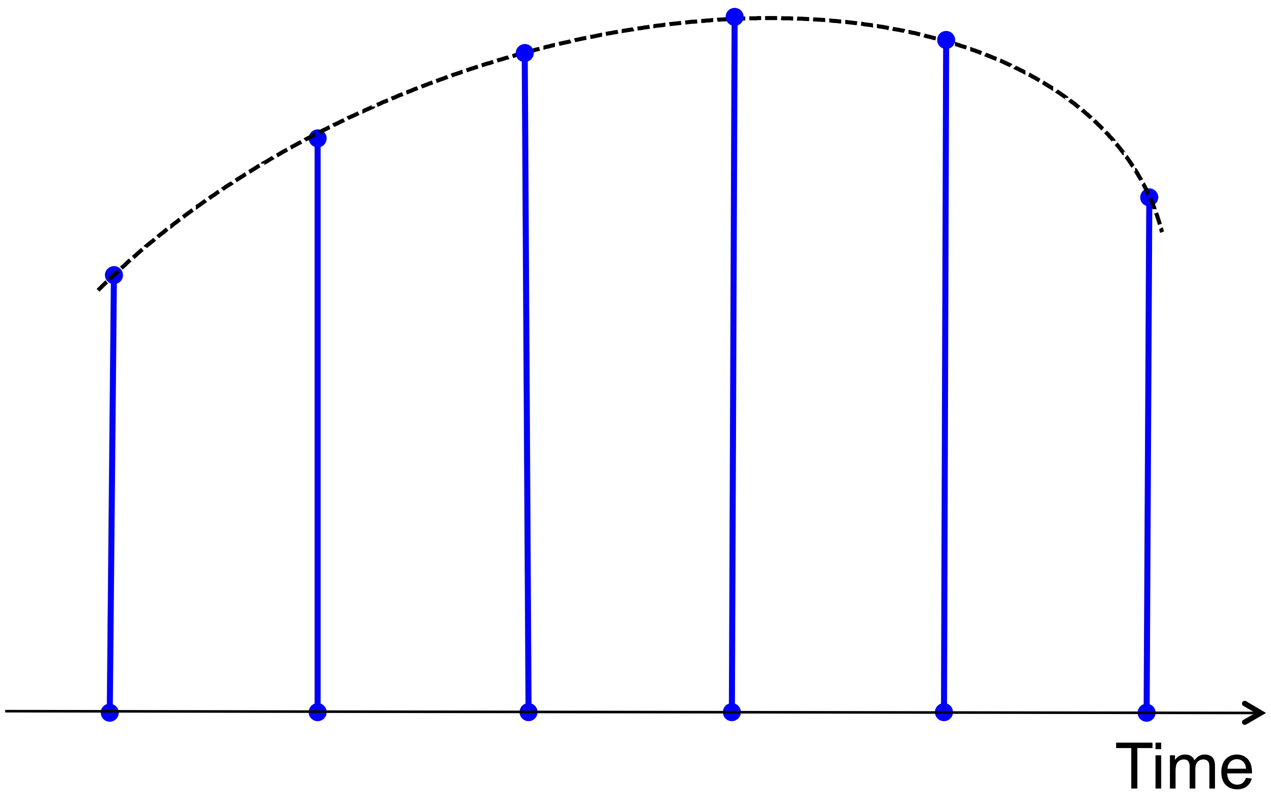

- How bad can it be?
  - Pretty bad it underestimates the peak 0.64 instead of 1
  - -3.9dB  of error!

# Why Values Between The Samples Matter!



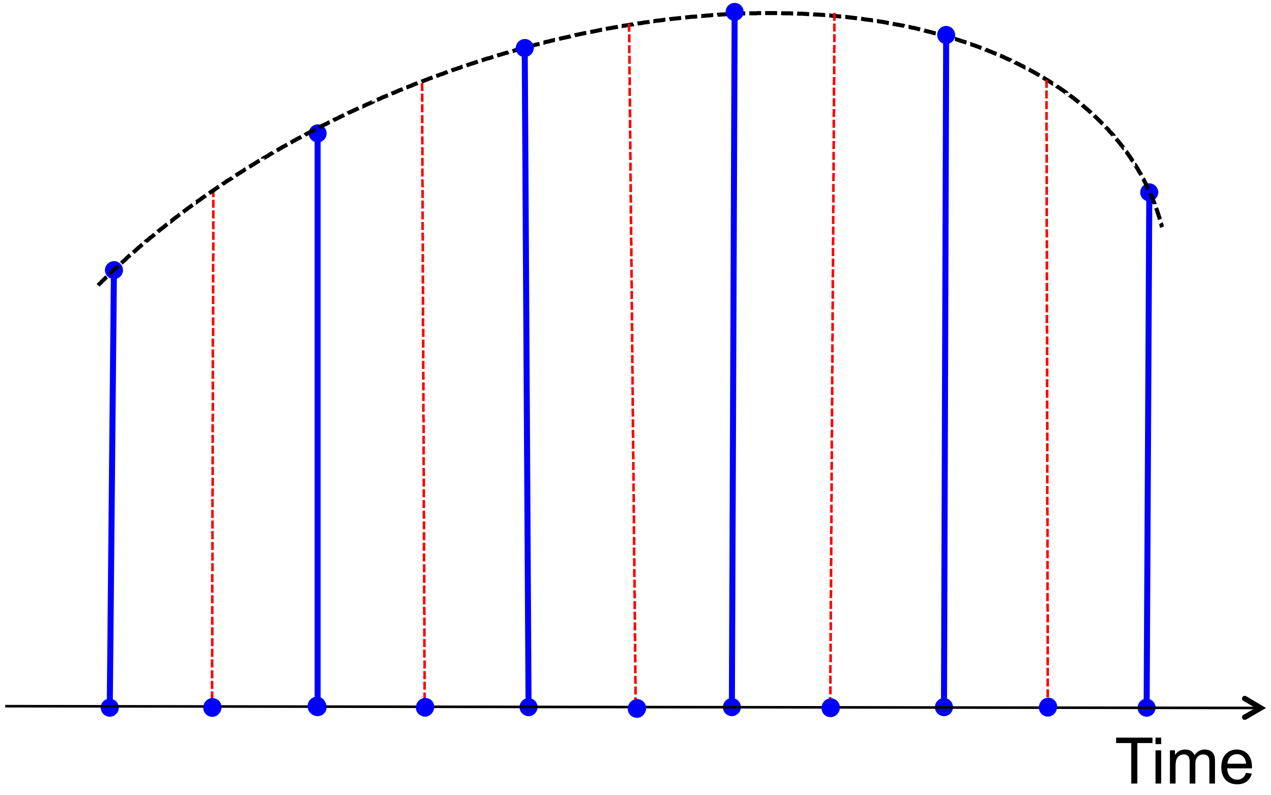

- Setting the maximum D/A output to the supply voltage
- Results in clipping downstream due to the waveform value between the samples
- Overshoot is approximately 1.5dB

# Estimating The Analogue Waveform

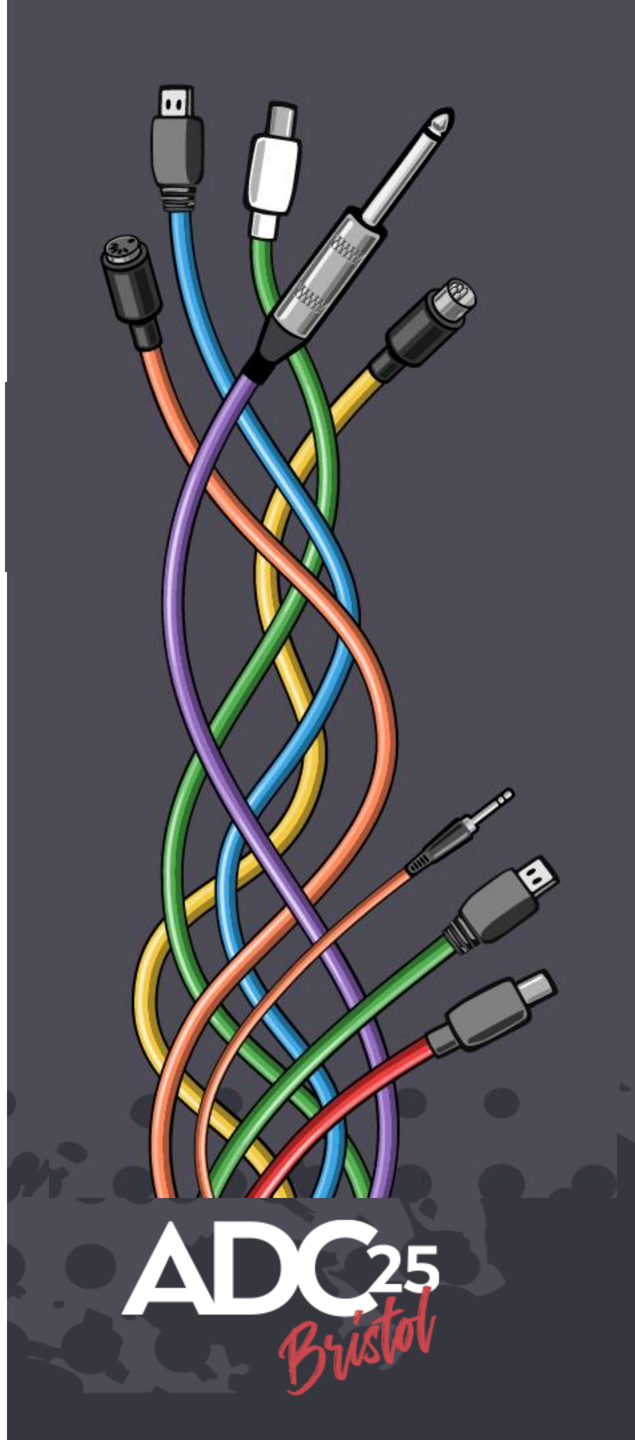



- One way is by increasing the sample rate

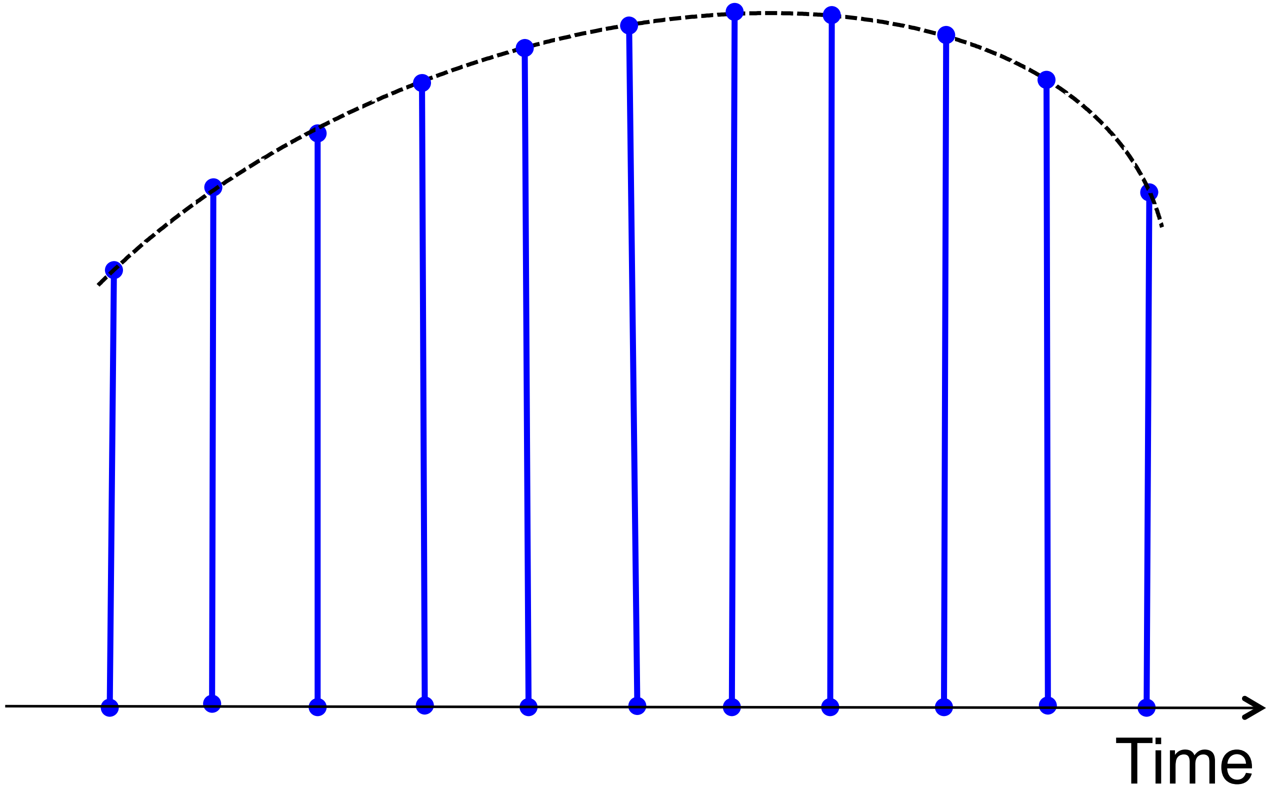
# Increasing The Sample Rate




- Insert zeros between samples
- And then filter to reconstruct the in-between samples

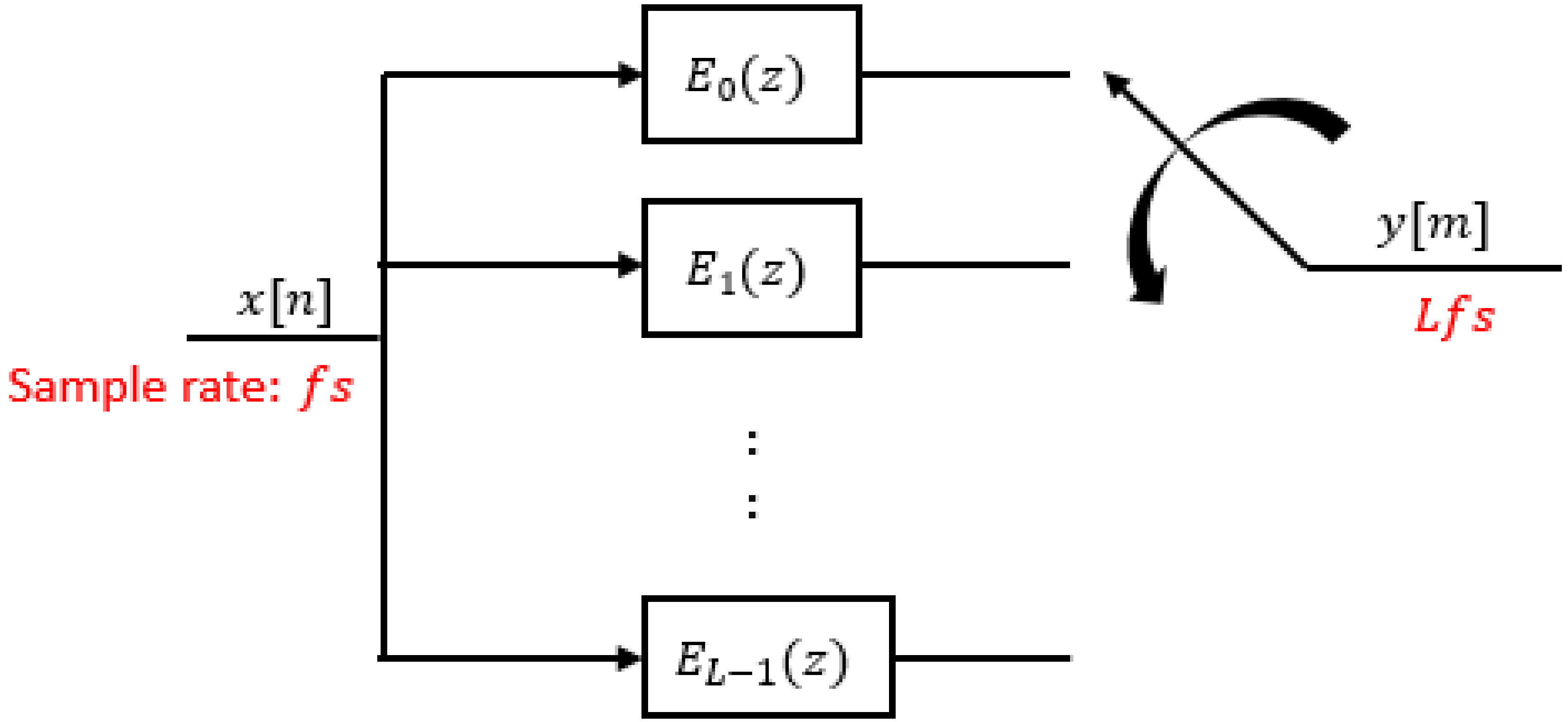


# Estimating The Analogue Waveform

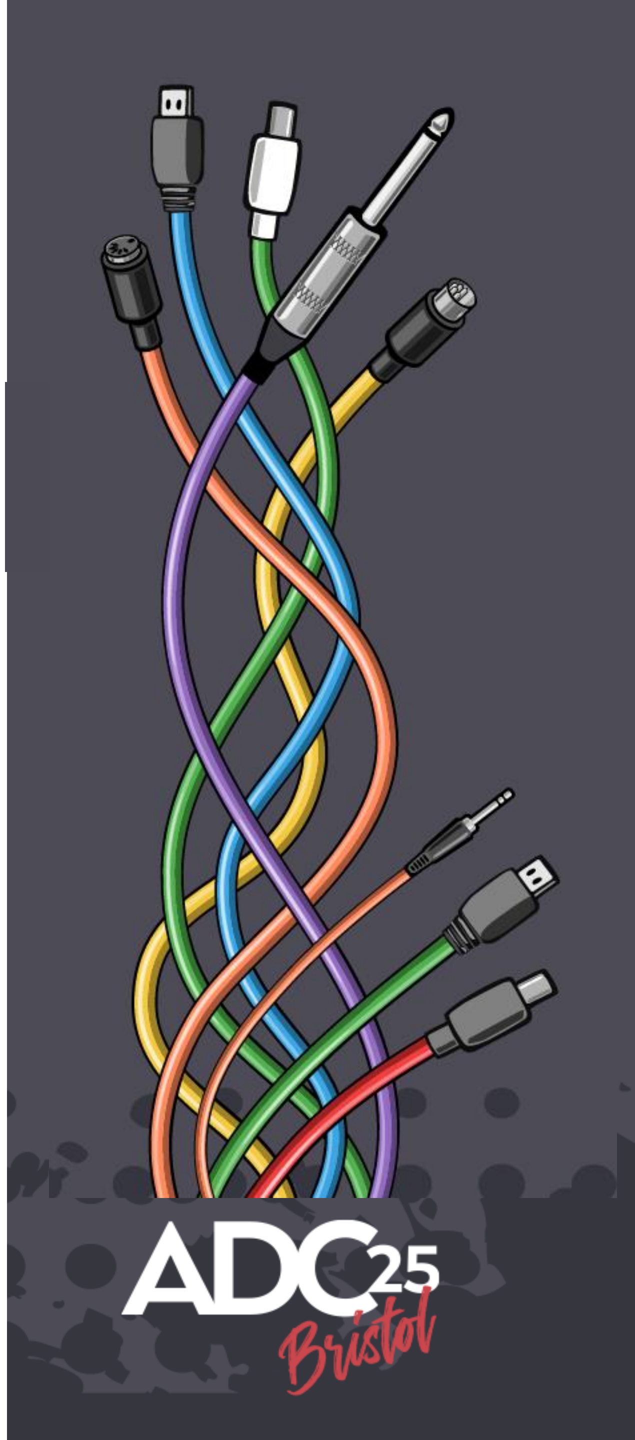



- 4 times up-sampling typically used
  - For Example, ITU BS 1770-4 ( Loudness meter)
  - But it introduces delay (6 samples)
  - And computation 48 multiply adds

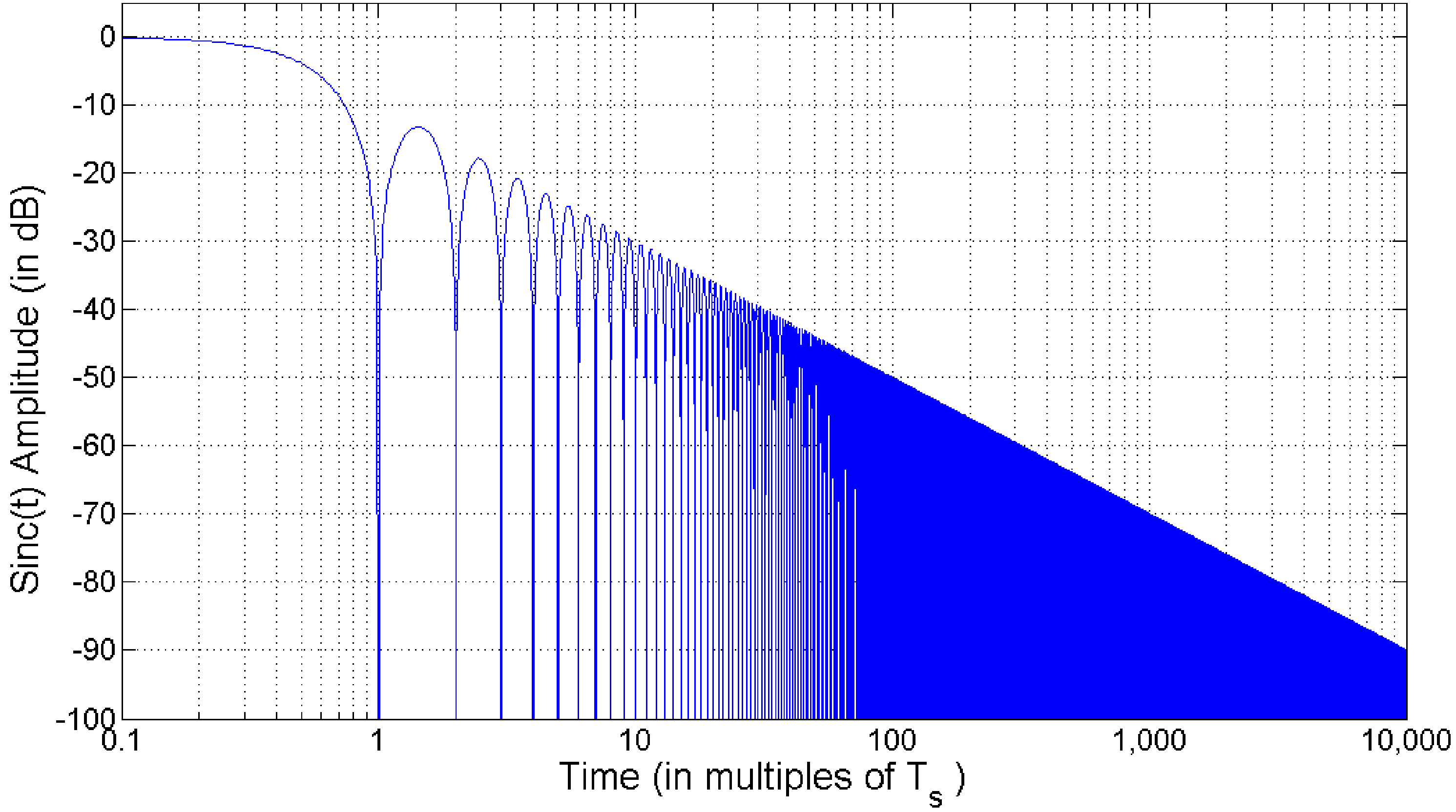
# Polyphase Interpolation By L



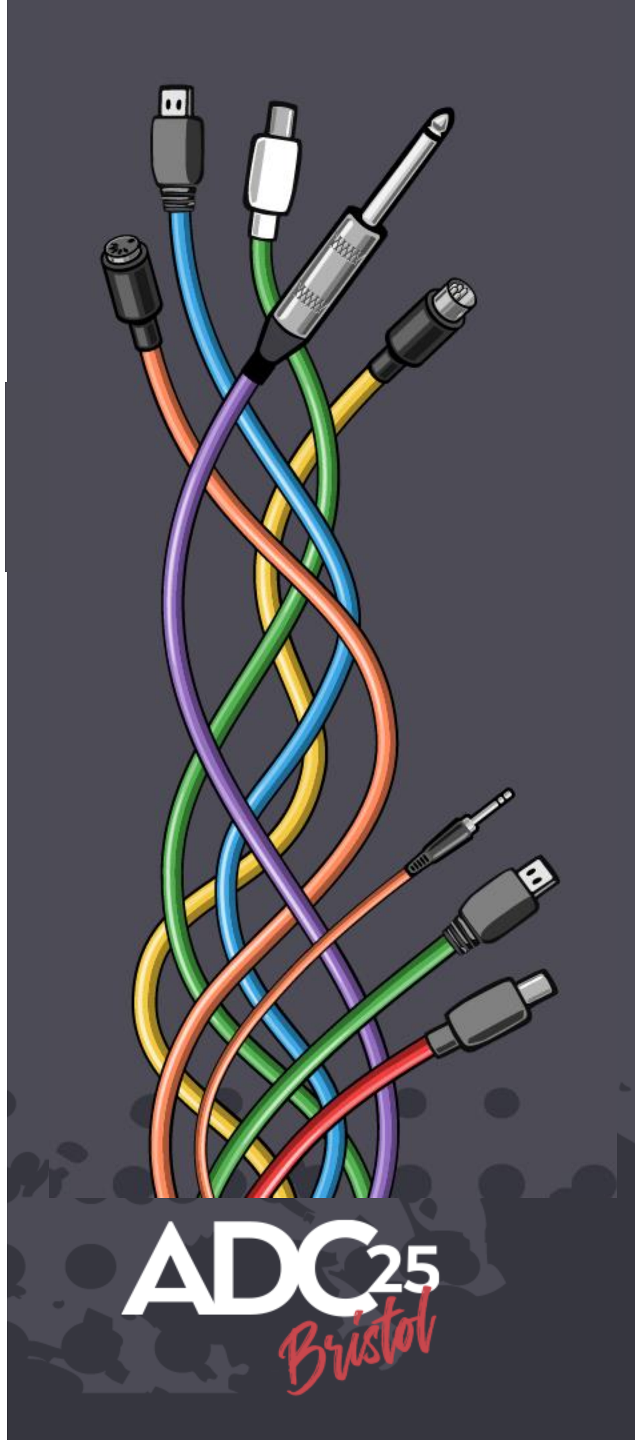

- Same input sample is fed to L low pass filters
  - Each calculates one intermediate sample position
  - By sub-sampling the Sinc coefficients at the higher sample rate
- The outputs are successively sampled
- To give L times interpolated output



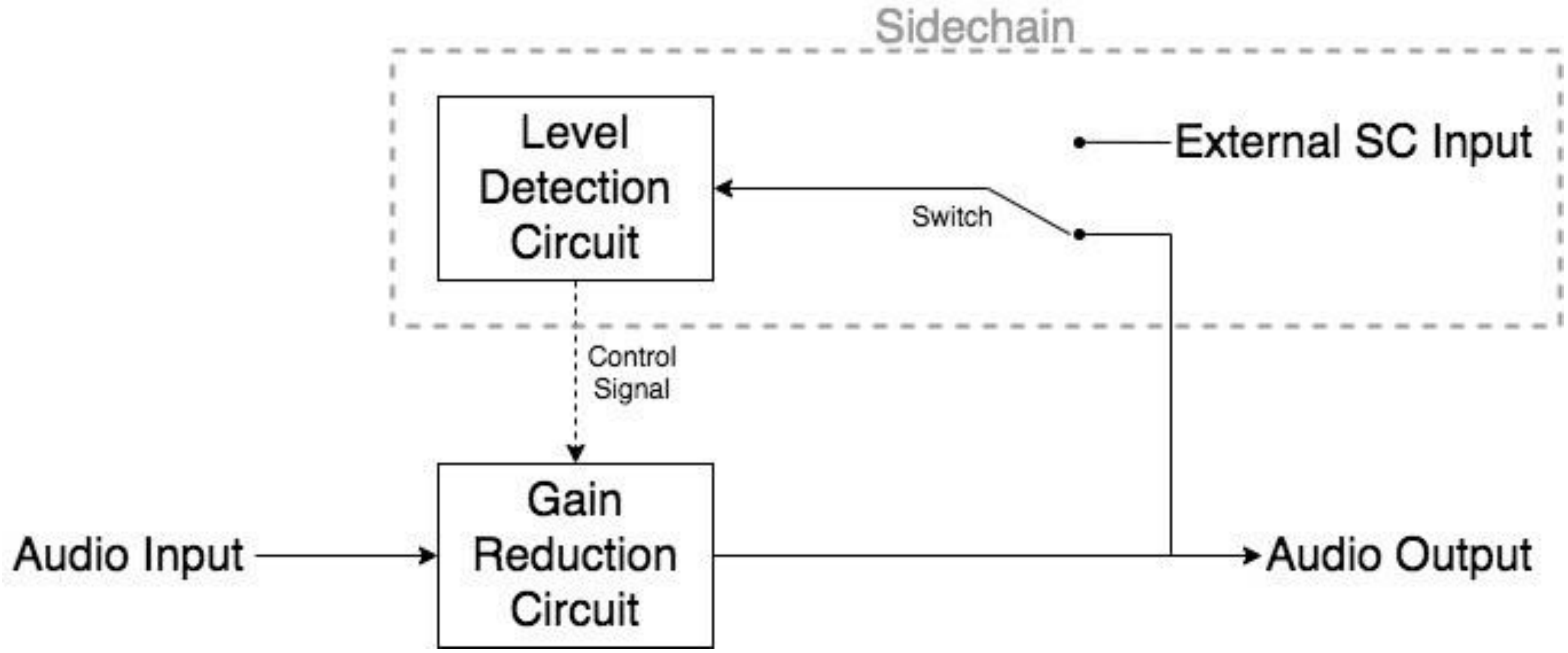
# What Happens When You Limit The Length?




- The longer the impulse the lower the estimation errors
  - But Sinc(t) decays very slowly ≈1/t)
  - So very long filters are required
- Problem with both delay and computation



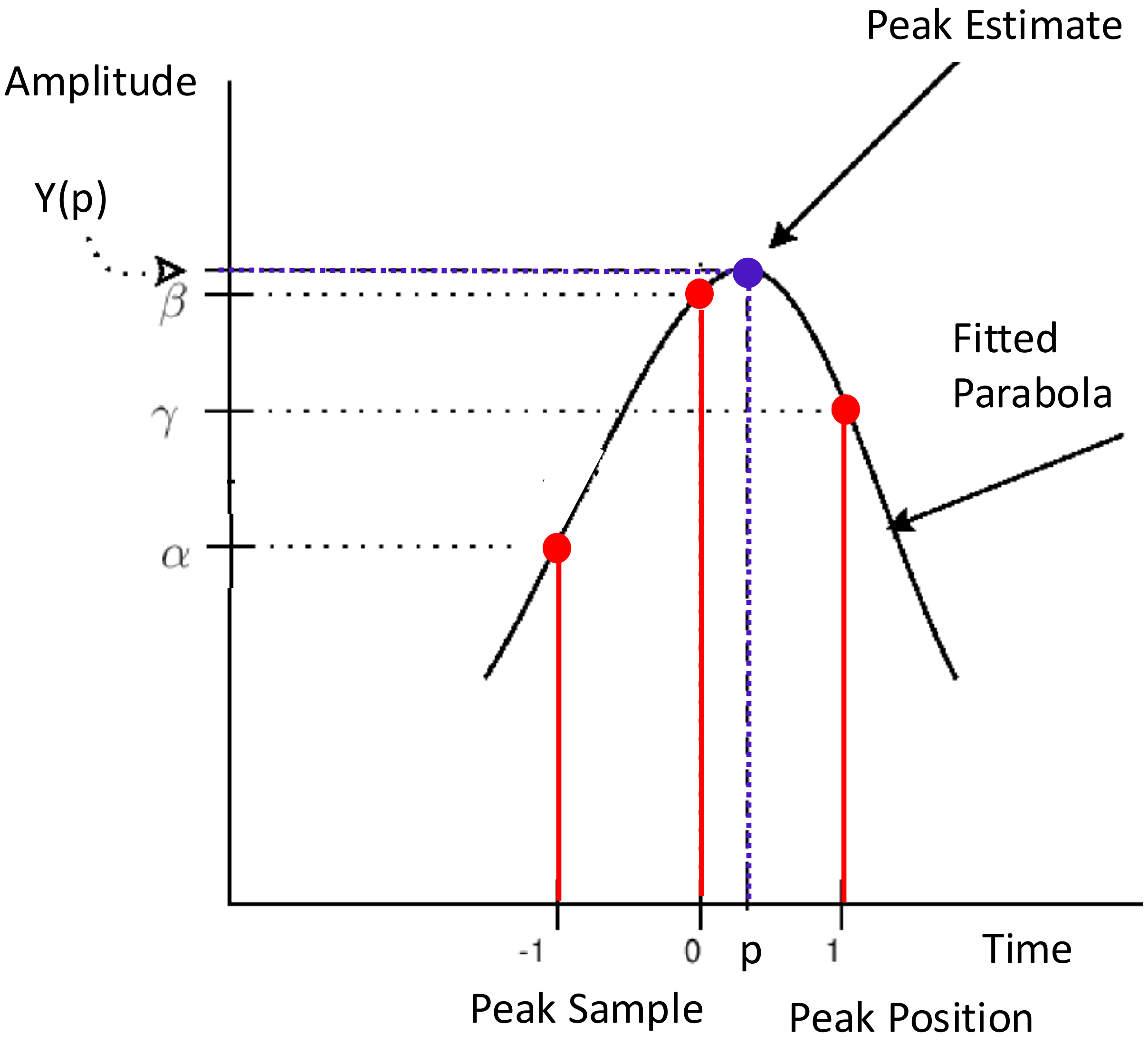
# The Need For Low Delay

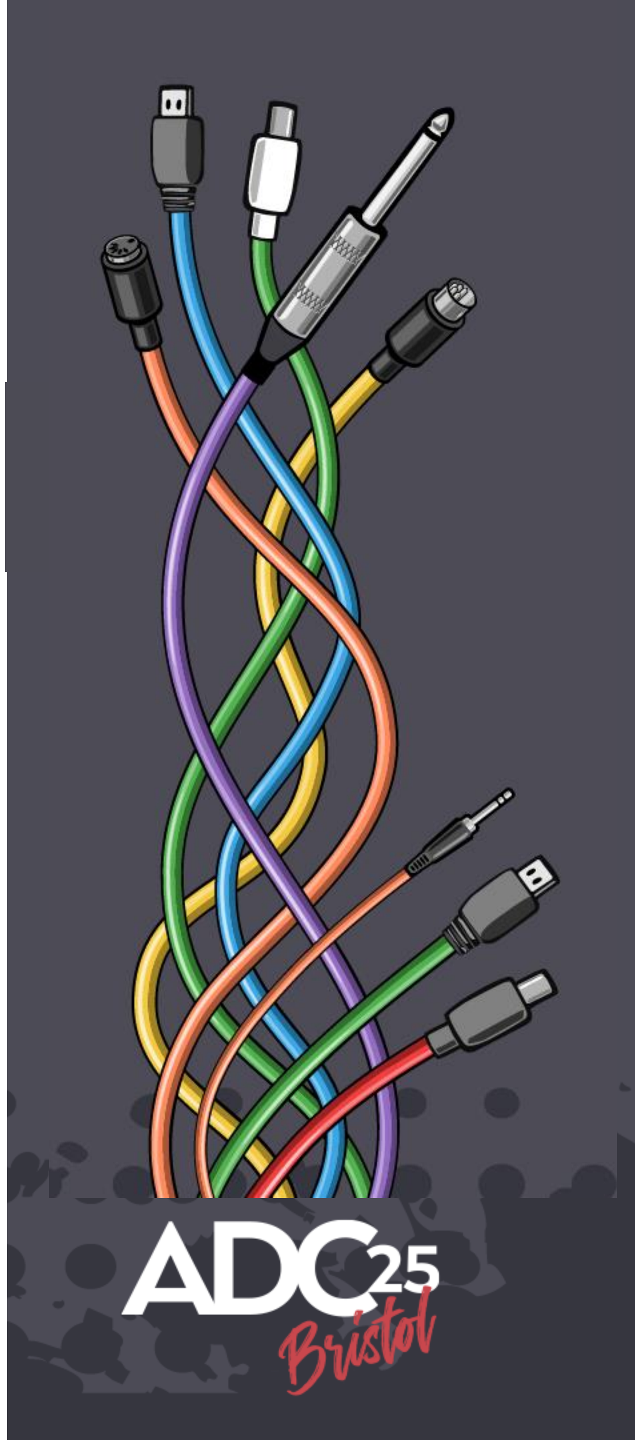



- Feedback Limiters/Compressors need a low delay level detection method
- Low delay can have lower computation

# Alternative 1: Quadratic Interpolation

- Use 3 adjacent samples
- Construct a parabola going through them
- Use the it to calculate:
  - ‘p’ the peak position
  - ‘Y(p)’ the peak height



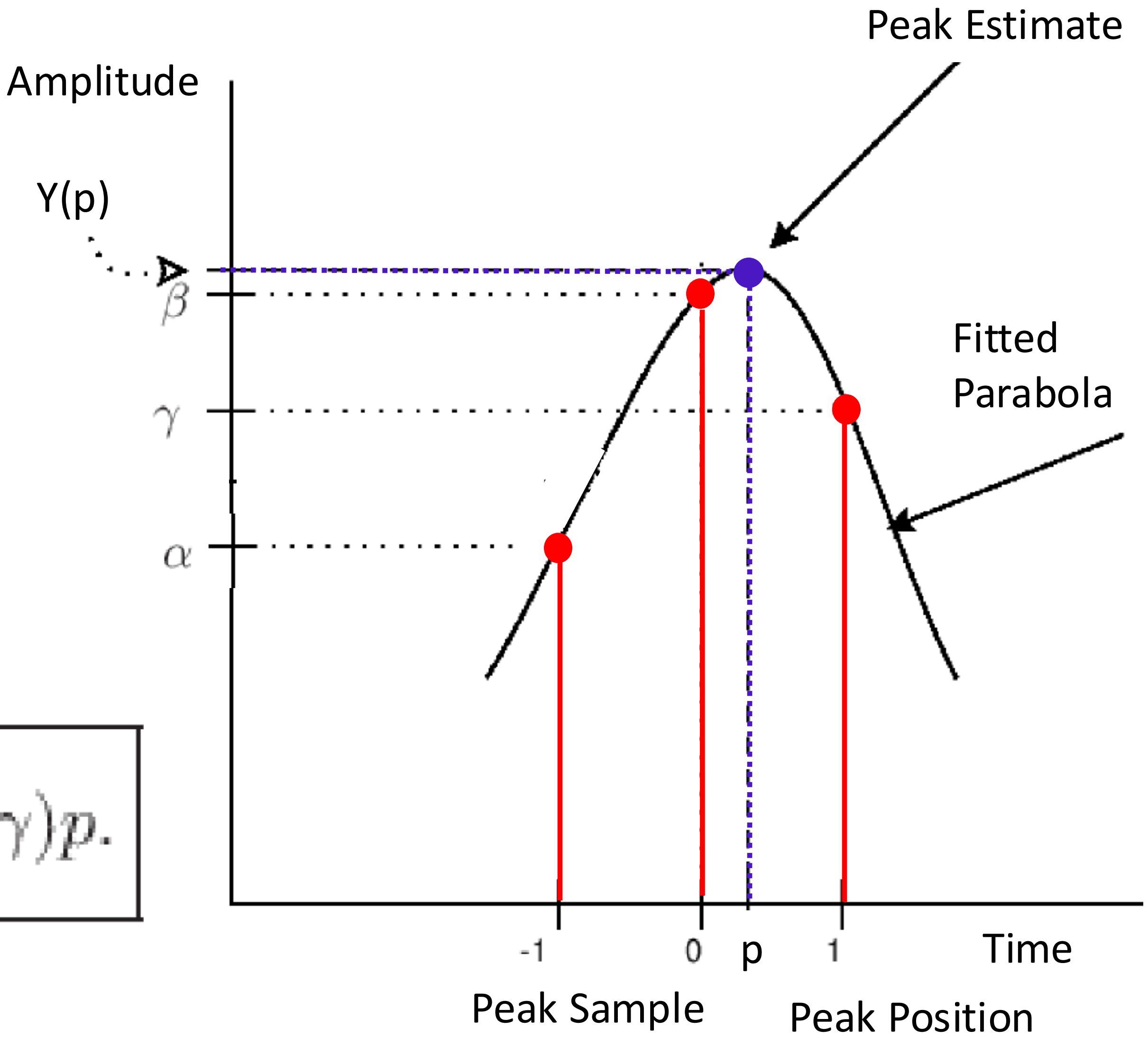



# Alternative 1: Quadratic Interpolation

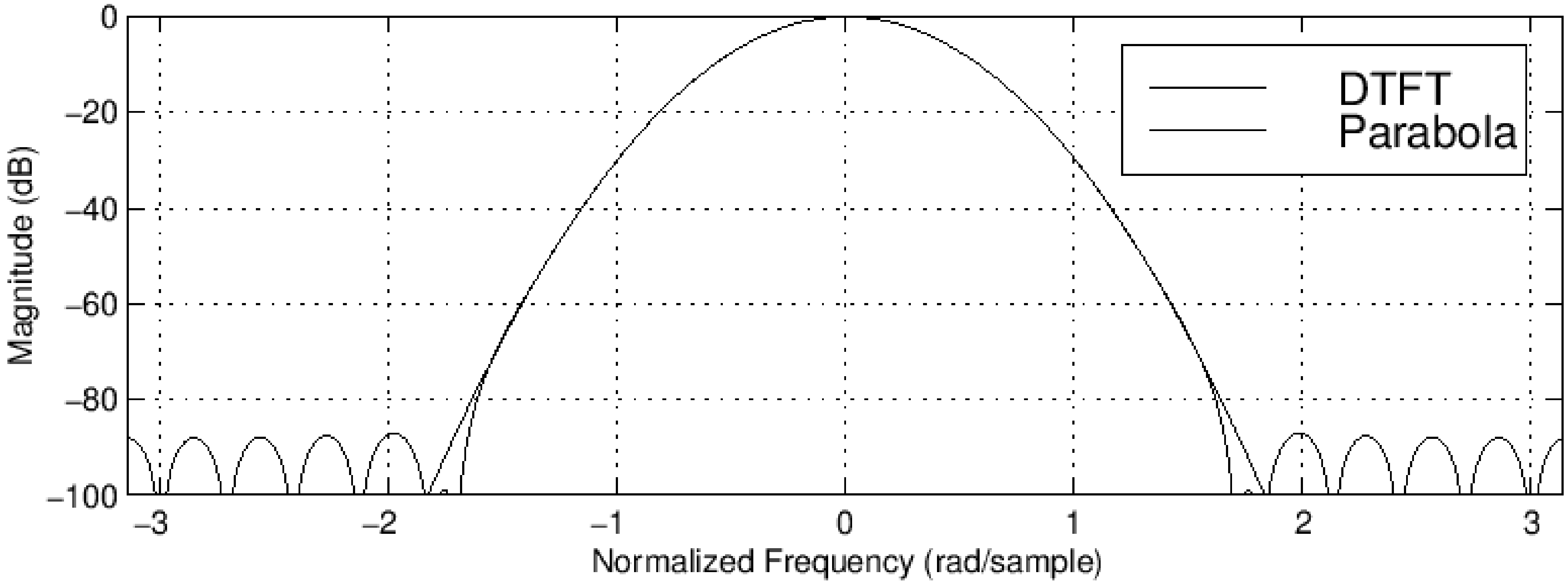
$$\boxed{p = \frac{1}{2}\frac{\alpha - \gamma}{\alpha - 2\beta + \gamma}}$$

$$\in [-1/2, 1/2]$$

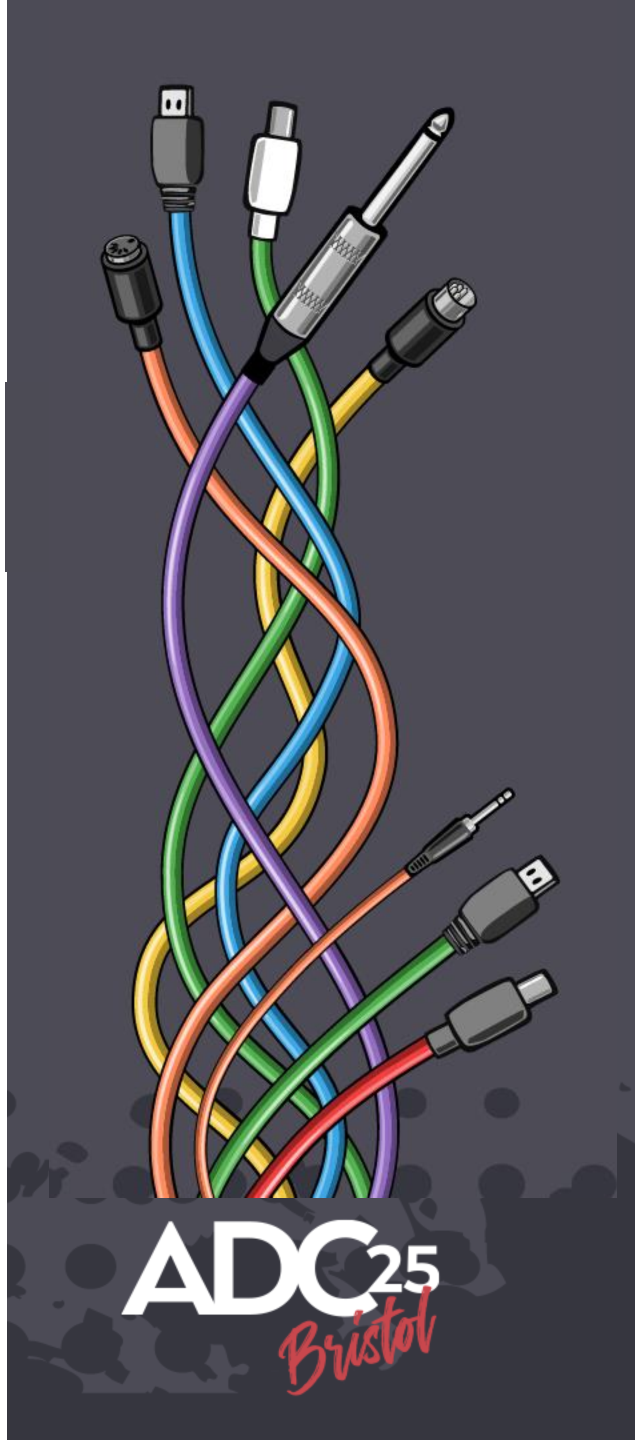
$$\boxed{y(p) = \beta - \frac{1}{4}(\alpha - \gamma)p.}$$



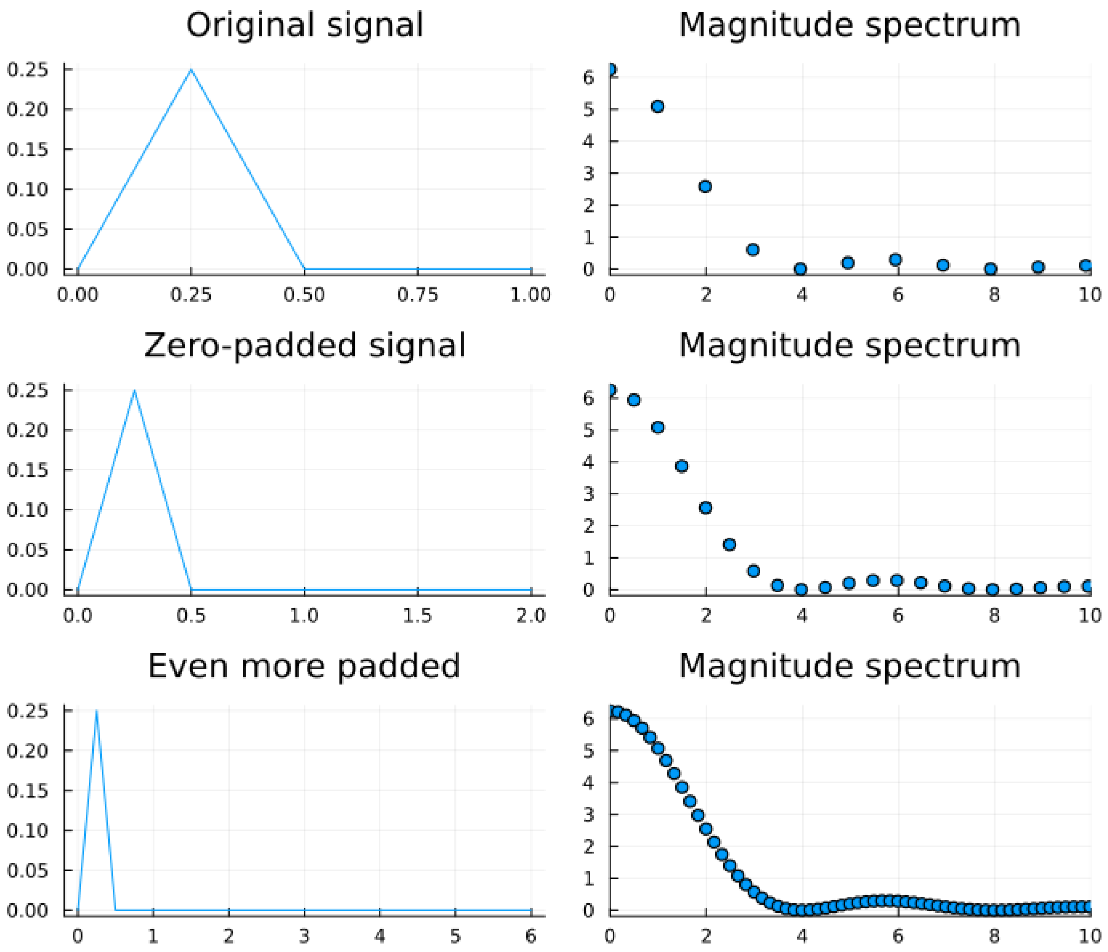
# Quadratic Interpolation: Frequency Peaks



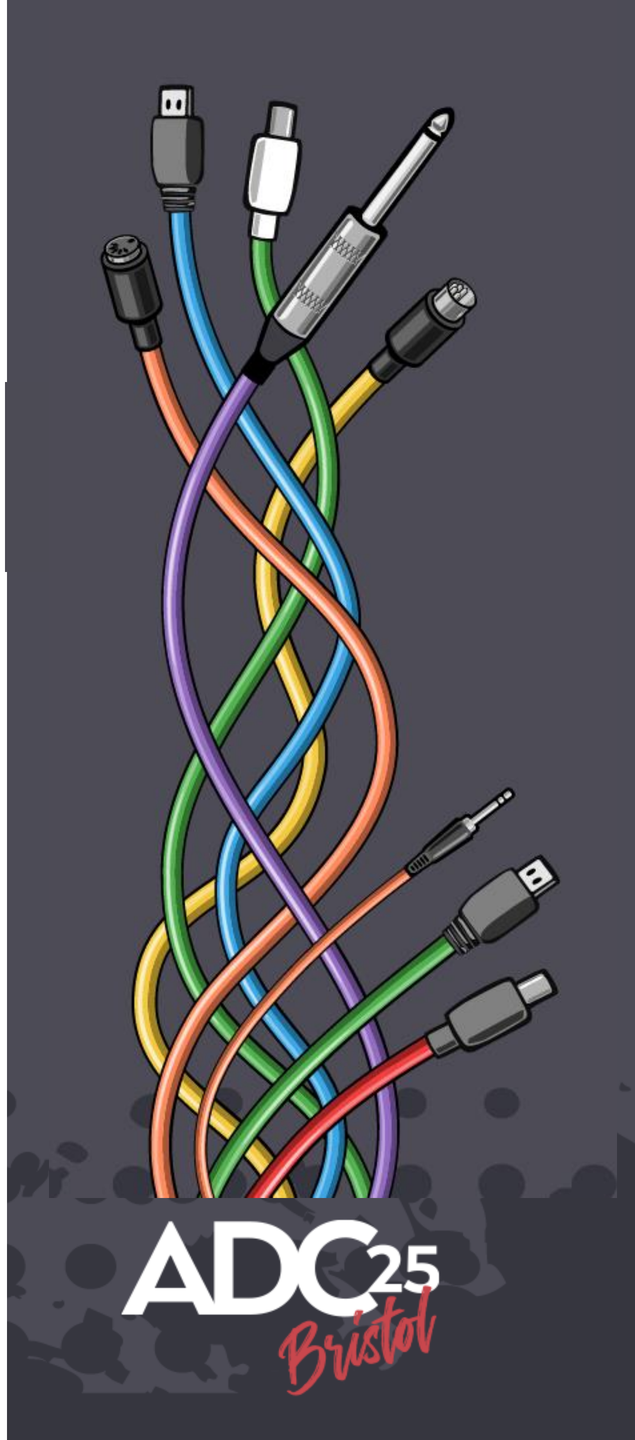

- Parabolic interpolation can be used for frequency domain peaks
  - Use Gaussian Window
    - Note can use Hamming trick to limit window length
    - Add 9% step, to cancel first bin



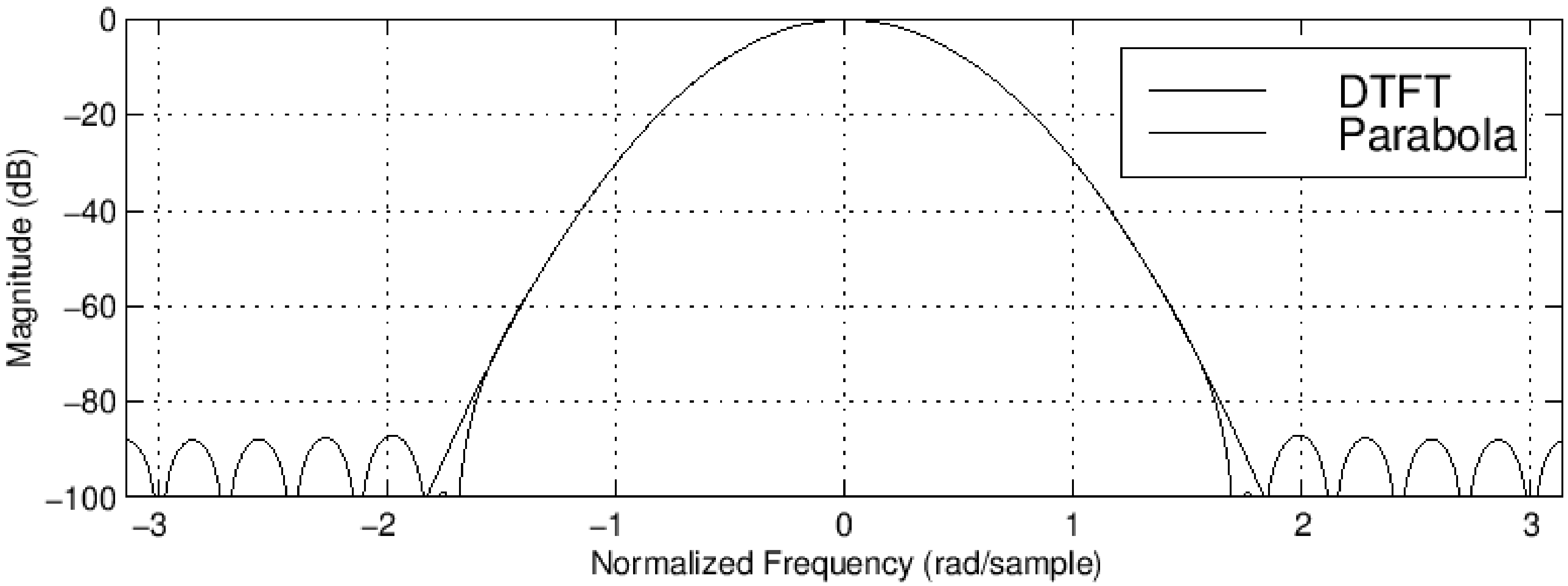
# Quadratic Interpolation: Frequency Peaks



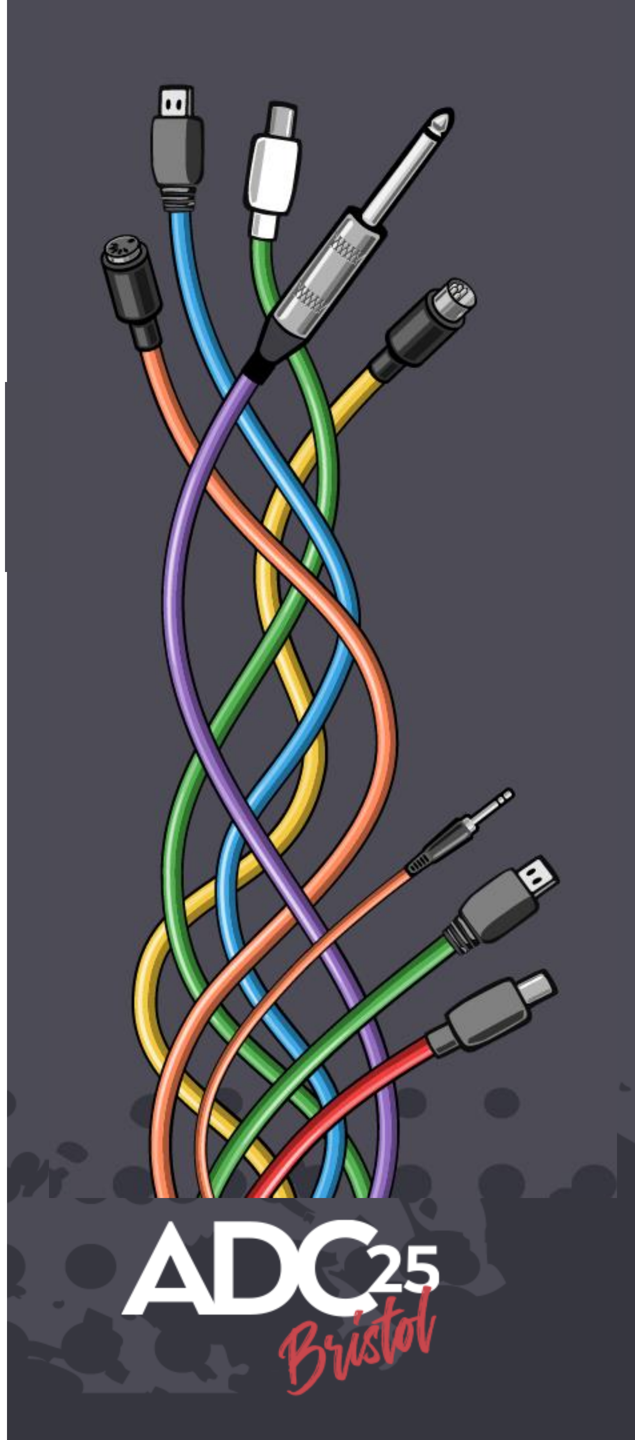

- Add zeros to interpolate the spectrum
  - At least 2 times
    - https://courses.grainger.illinois.edu/bioe205/sp2023/lectures/lec09/



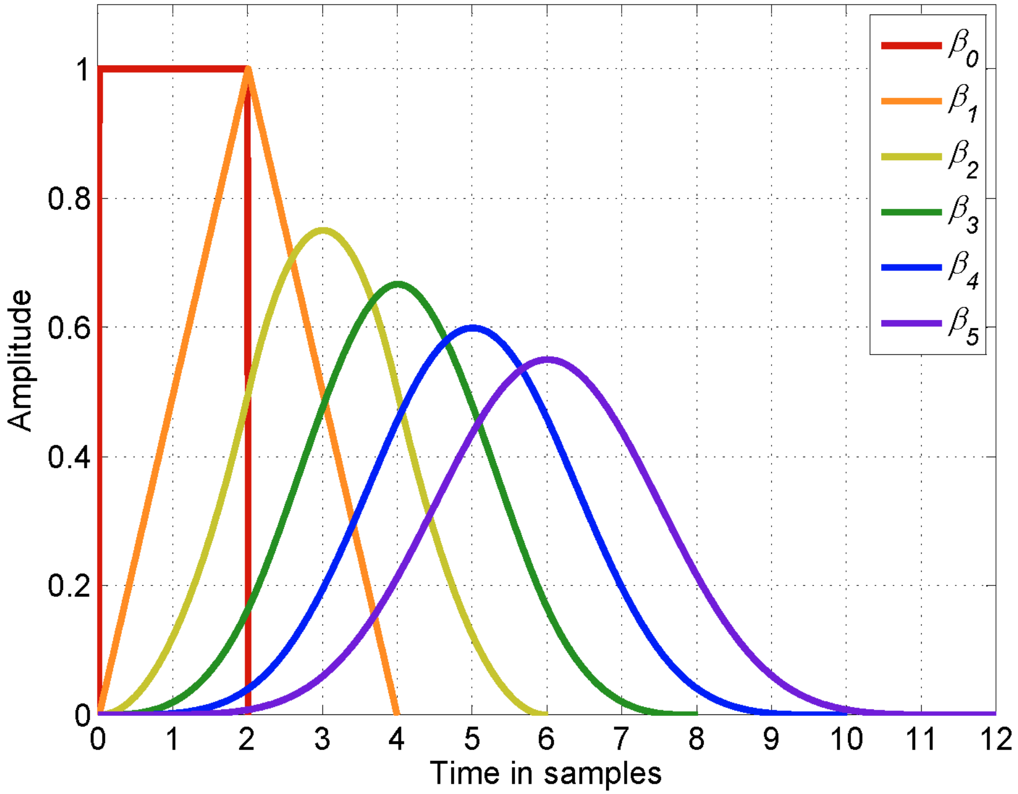
# Quadratic Interpolation: Frequency Peaks




- Parabolic interpolation can be used for frequency domain peaks
  - Use Parabolic interpolation on the log-magnitude result
  - Because the lobes shape parabolic at the top
    - From Julius Smith of CCRMA

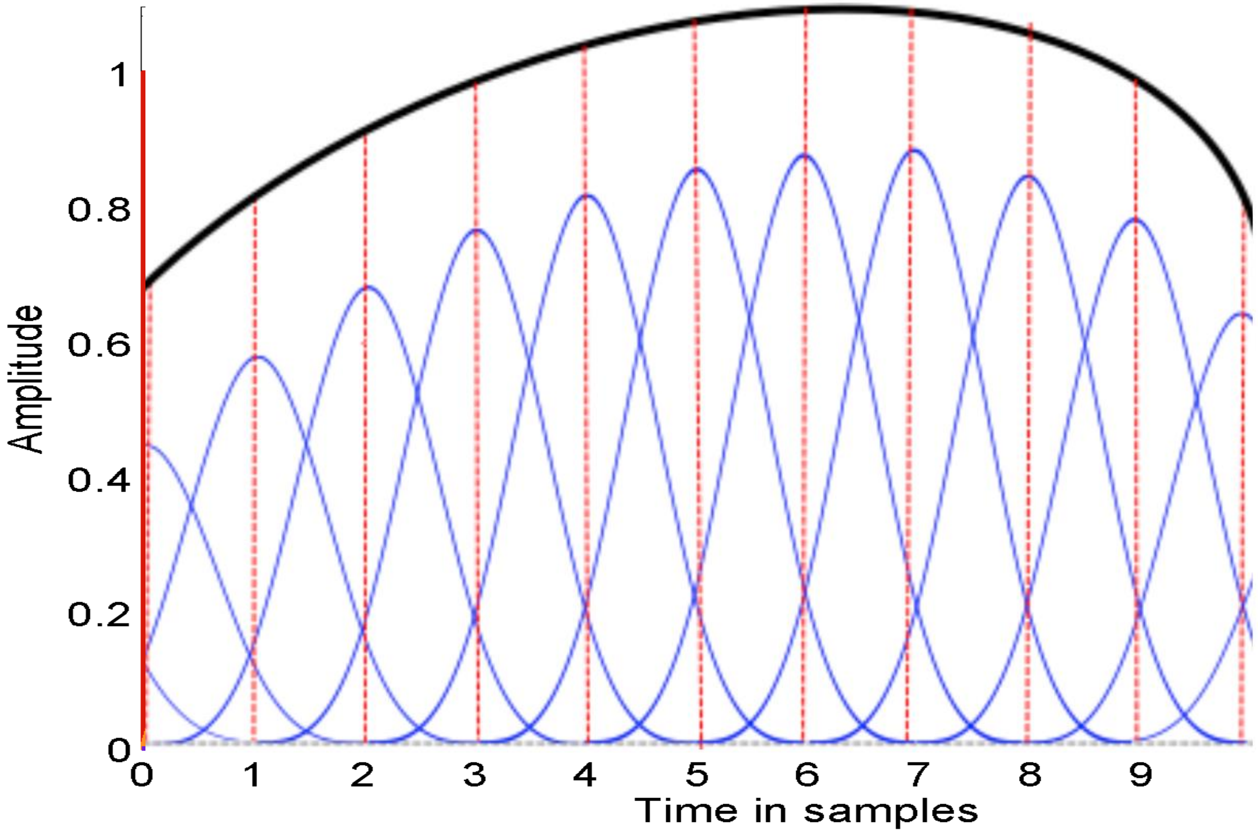


# Alternative 2: β-Spline Interpolation



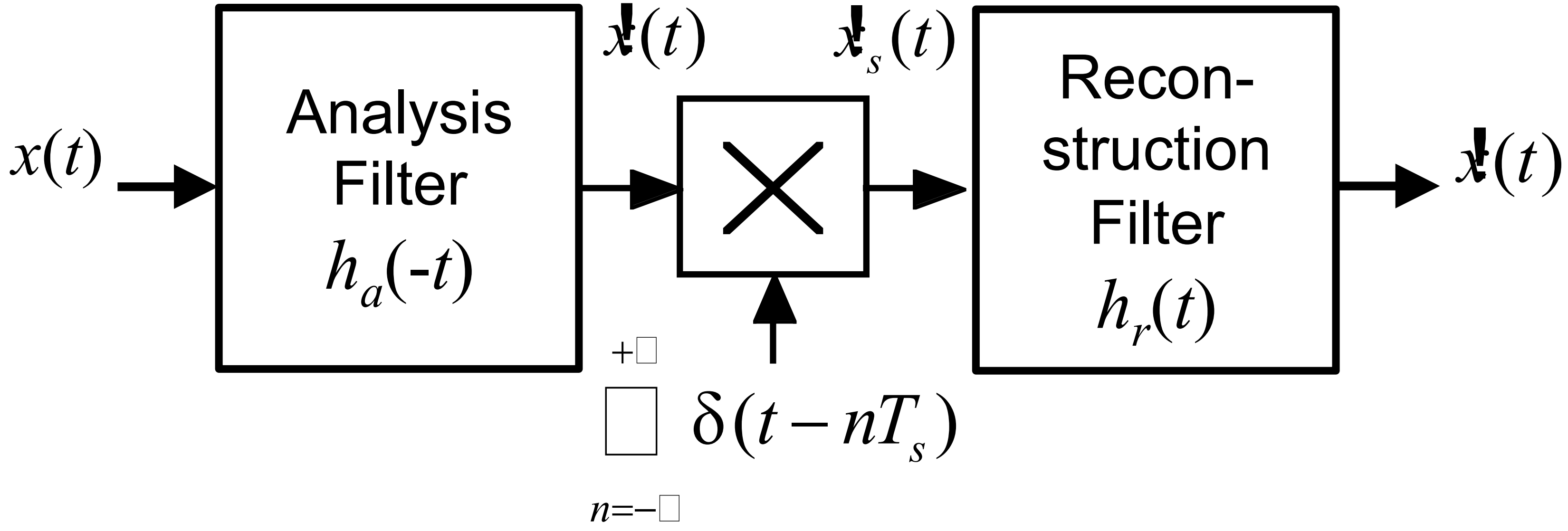

- They are formed by successively convolving the
  - Zeroth order spline, a simple rectangle of width one, $\beta_0$
- To yield the higher order $\beta_n$ splines
  - $B_1$ is equivalent to linear interpolation
- They have the advantage of a finite time extent, not infinity!

# Alternative 2: β-Spline Interpolation



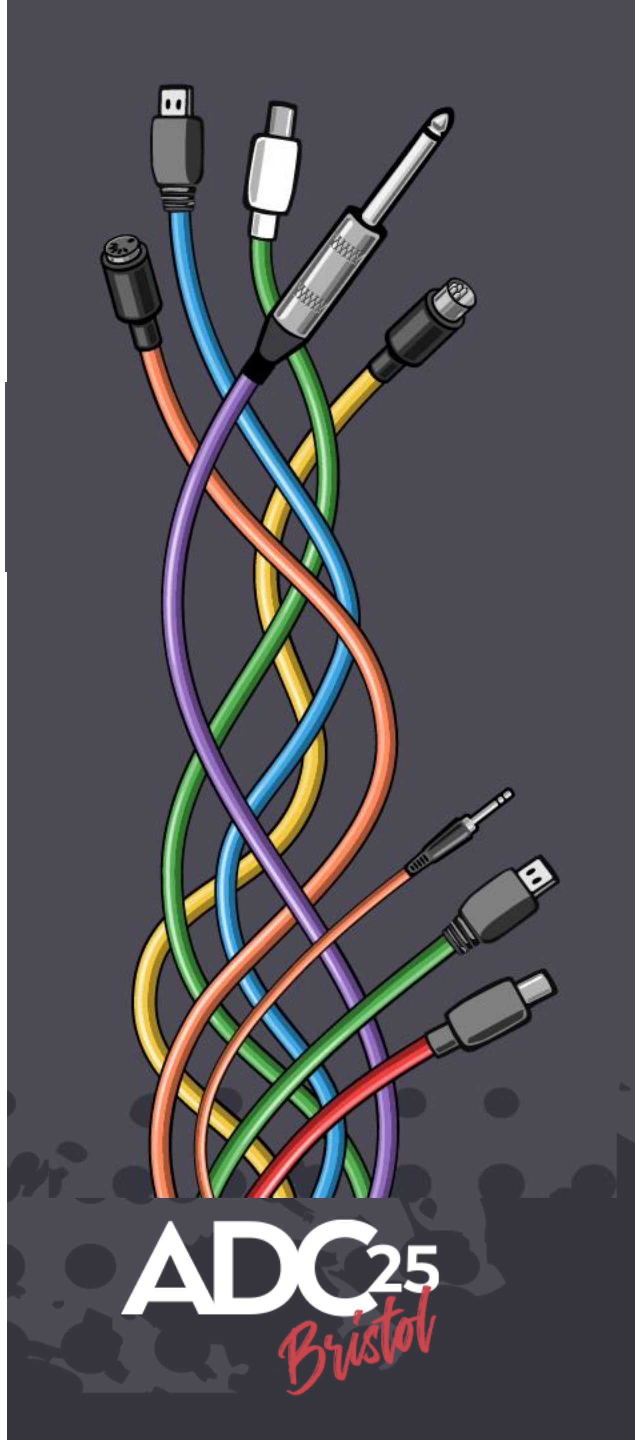

- Unfortunately, the splines overlap other samples
  - They are not independent
- So, you can’t use the sample values directly
  - Because they interact.
- Usually requires a matrix inversion per output point

# Using a Filter Instead Of Matrix Inversion

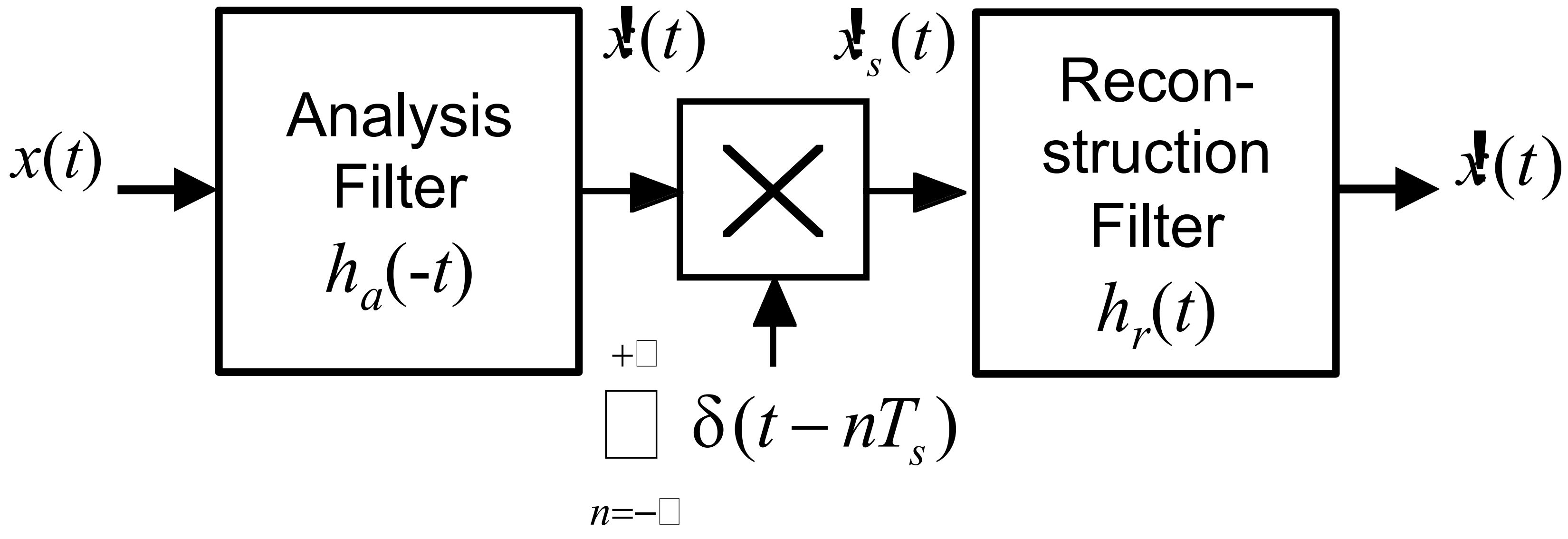



- Unser showed you can replace the matrix inversion by a filter.
  - With a transfer function *H(z)* of:

$$H(z) = \frac{6}{z + 6 + z^{-1}}$$



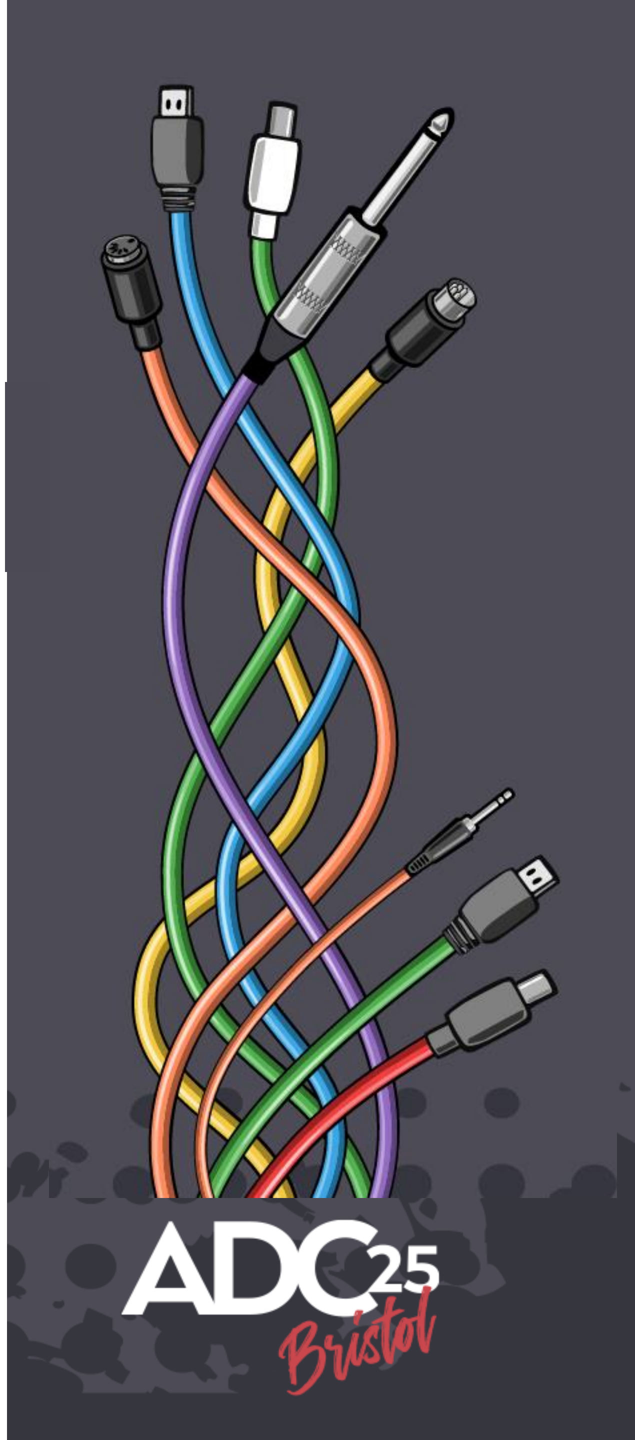
# Using a Filter Instead Of Matrix Inversion




- Unser showed you can replace the matrix inversion by a filter.
  - With a transfer function *H(z)* of:

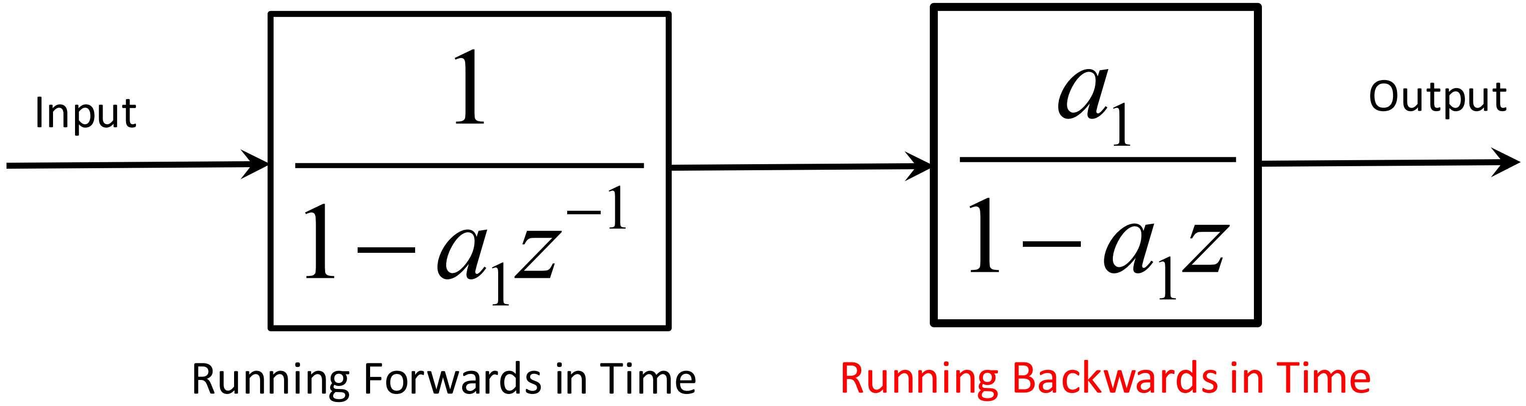
$$H(z) = \frac{6}{z + 6 + z^{-1}}$$

- Unfortunately, it's non-causal!
  - (Pre-cognitive processing anyone? )

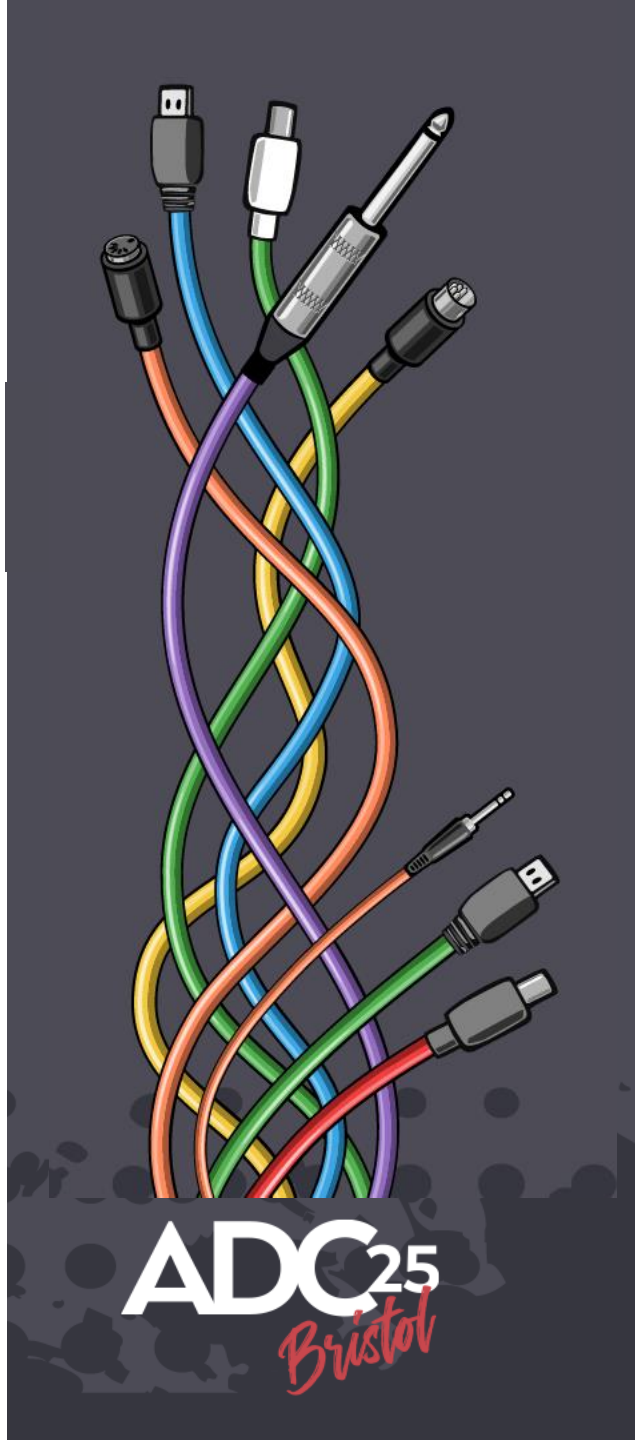


# Using a Filter Instead Of Matrix Inversion

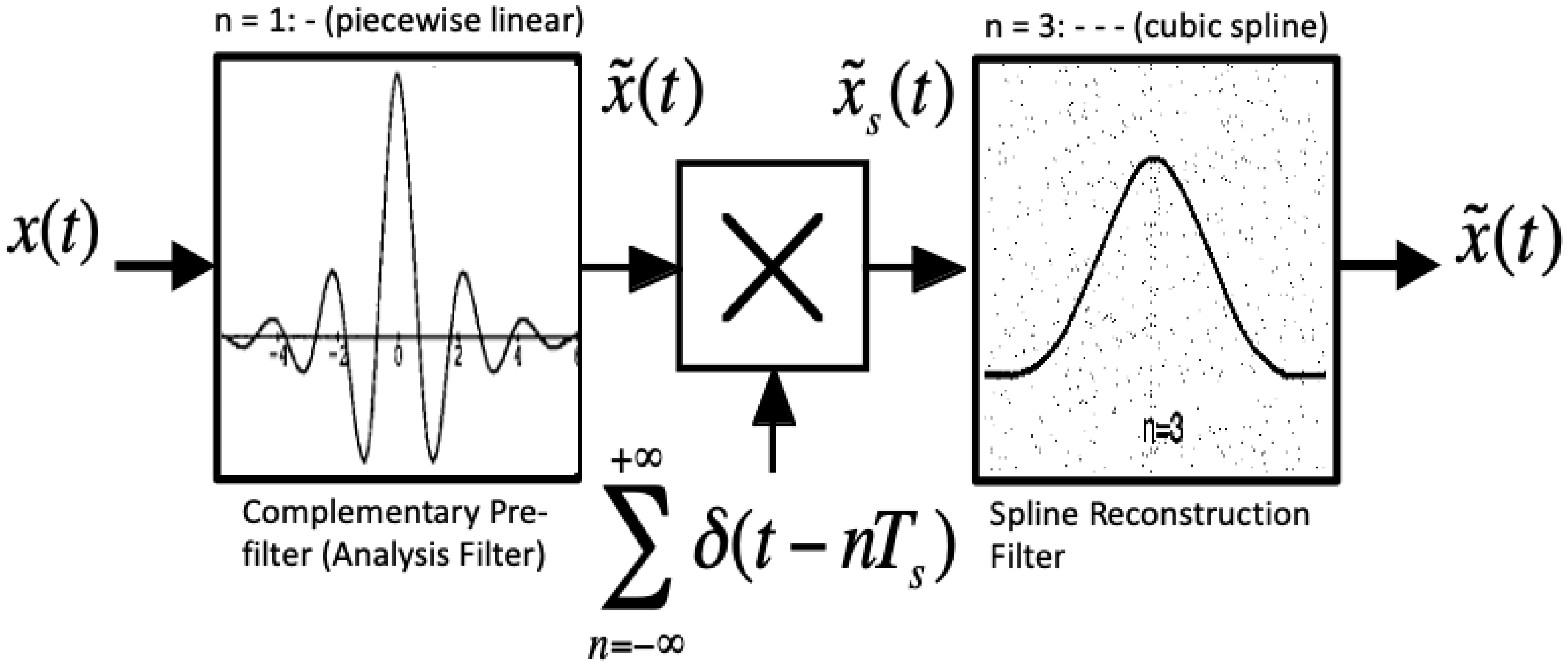
Complementary Pre-filter for a 3rd order spline $a_1 = \sqrt{3} - 2 = -0.26794919243$




- Fortunately, this can be realised using two IIR filters.
  - One running forwards in time
  - And one running backwards in time!
- Tricky to do in real time
  - But ok for mastering and pictures.
- In practice one must make a windowed FIR equivalent
  - Shorter than Sinc($t$) because it decays faster (exponentially vs $1/t$).

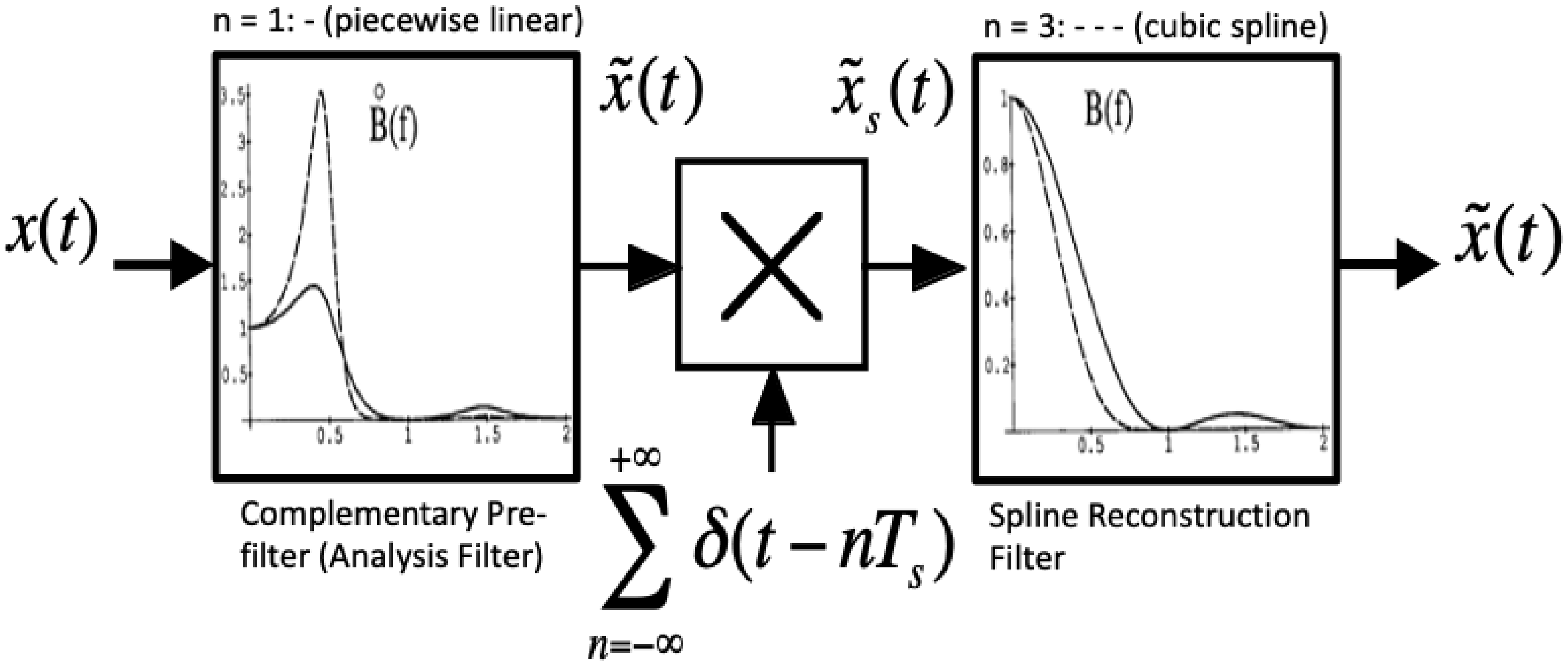


# Using a Filter Instead of Matrix Inversion

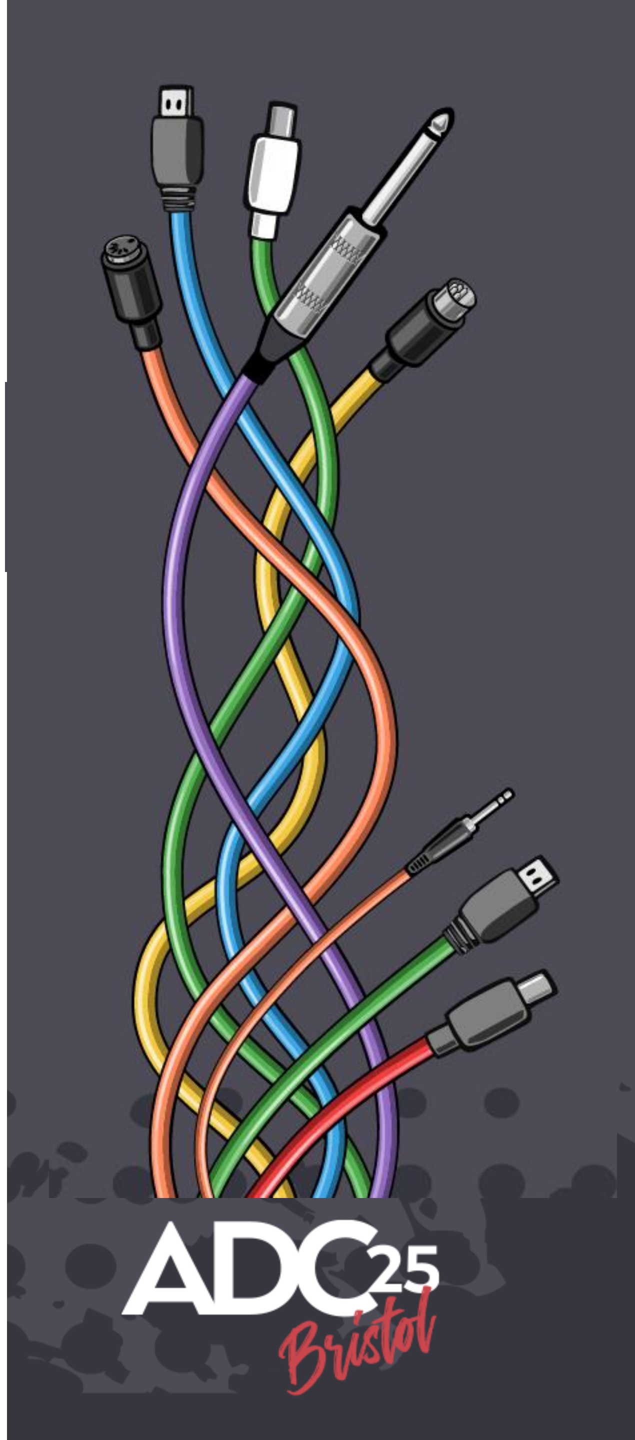



- This is what the filters look like in the time domain, for n=3
  - Reconstruction filter has a finite impulse response.
  - Pre-filter has an infinite impulse response so is truncated (windowed)

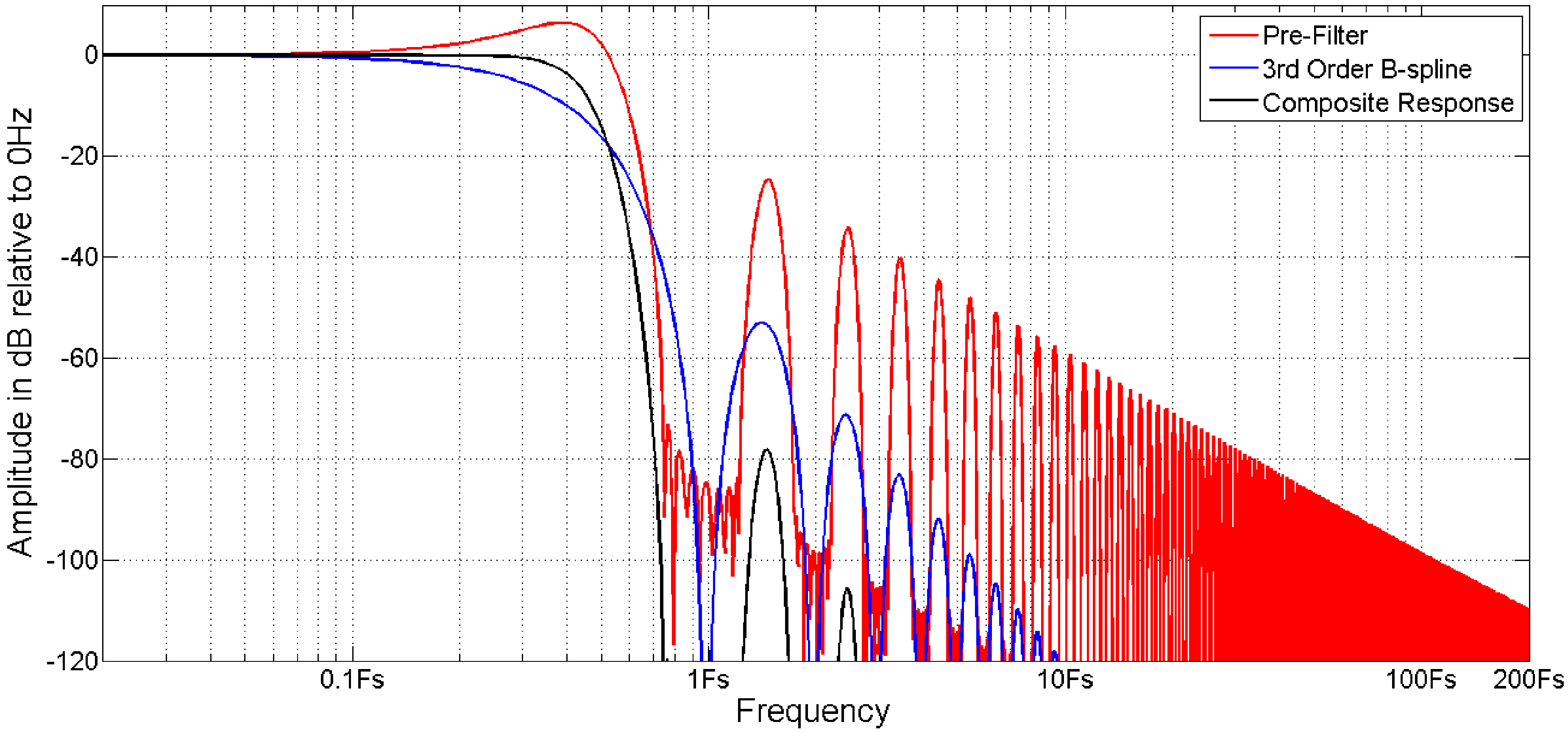
# Using a Filter Instead of Matrix Inversion



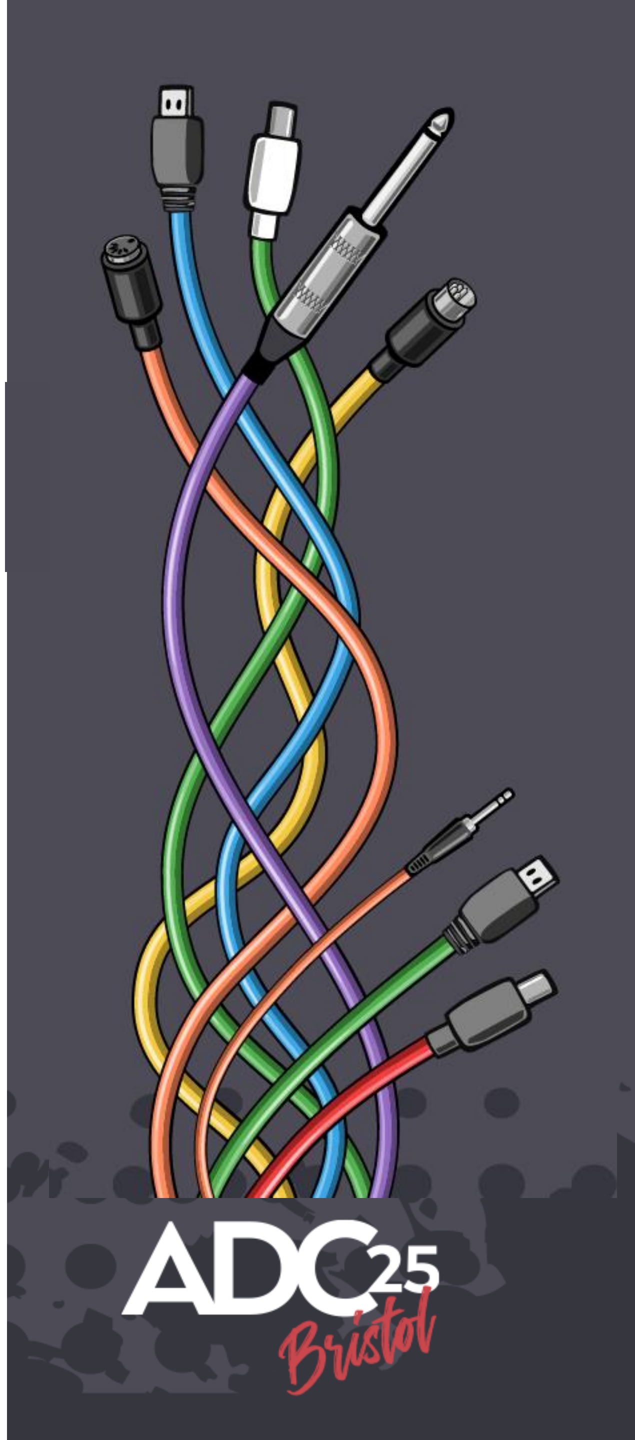

- This is what the filters look like in the Frequency domain, for n=3
  - Reconstruction filter has a casual roll-off response.
  - Pre-filter has a peaked response



# Using a Filter Instead of Matrix Inversion

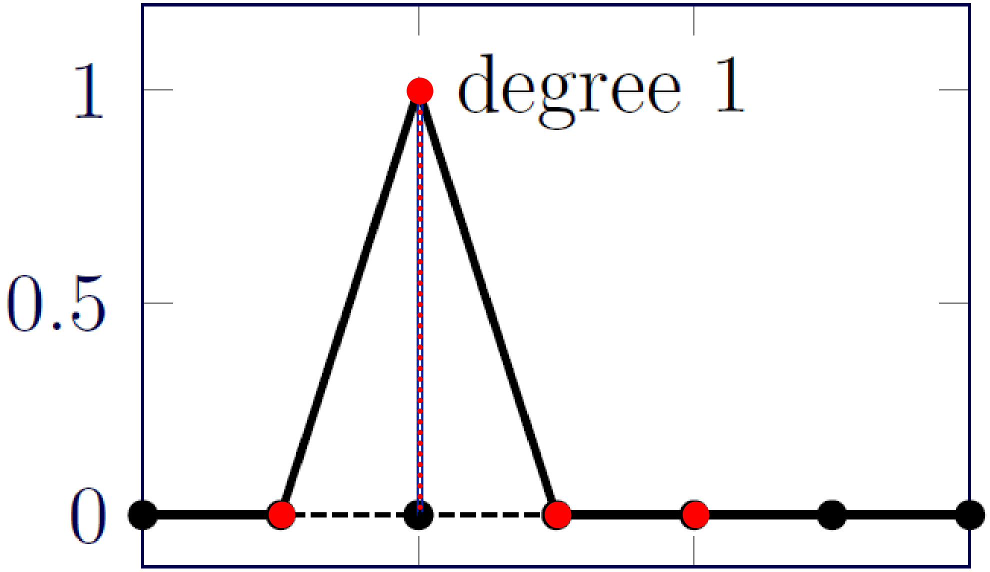


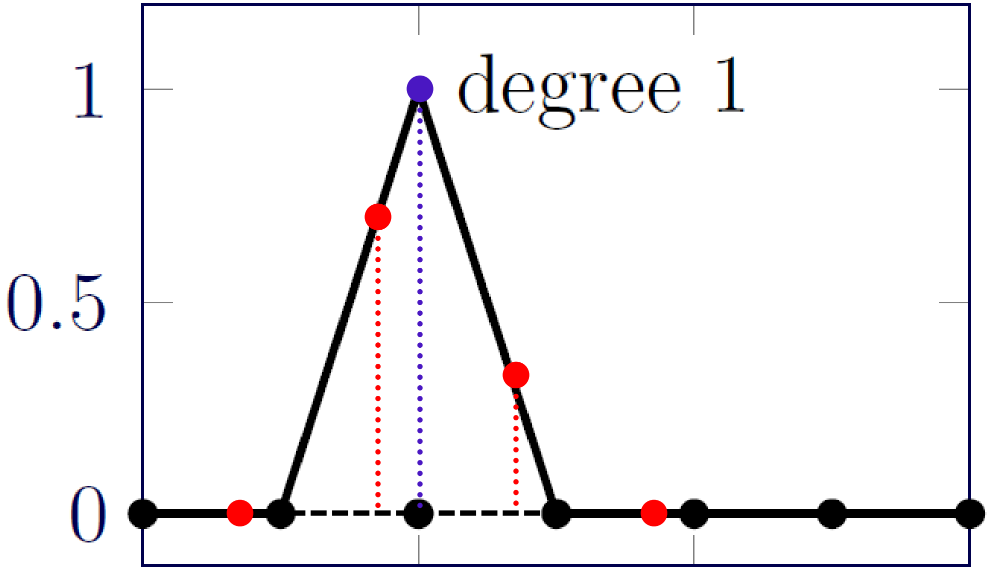

- This is what the filters look like in the Frequency domain, for n=3
  - Spline reconstruction filter has a casual roll-off response.
  - Pre-filter has a peaked response

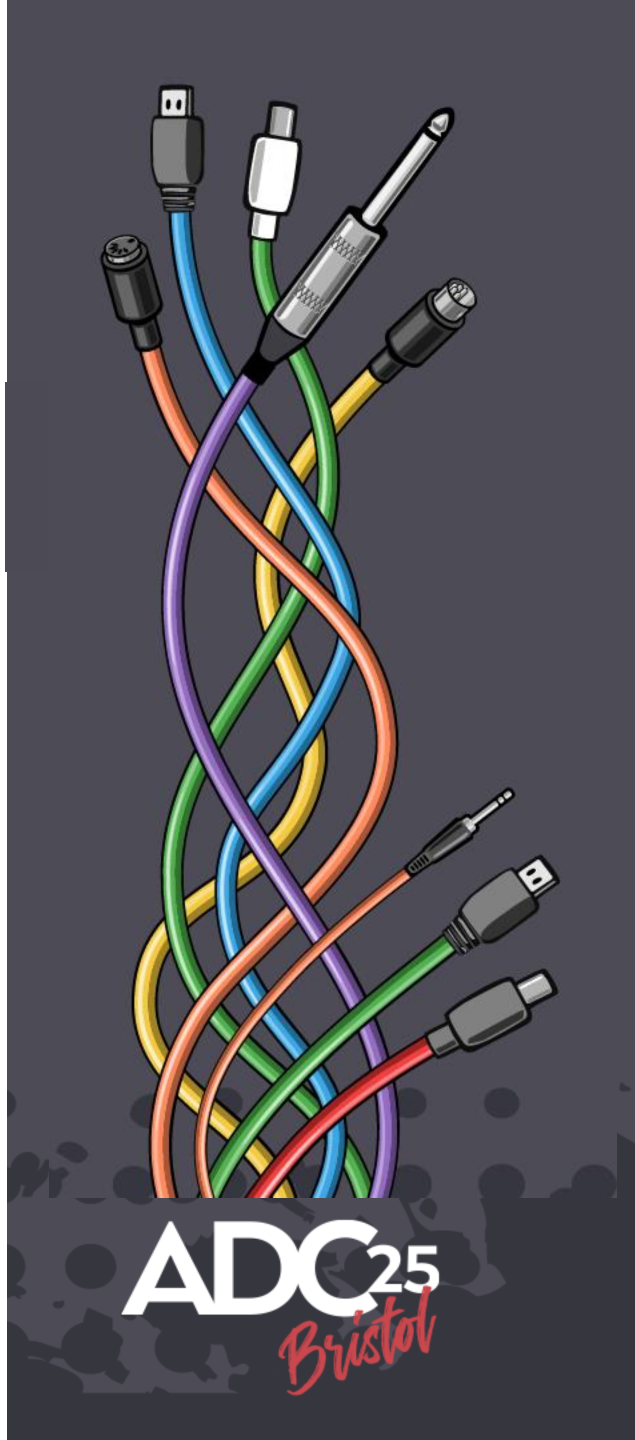


# Filter Based $\beta_1$-Spline Interpolation

- When the $\beta_1$ spline is on a sample point
  - It outputs that sample value
- When the $\beta_1$ spline is between two sample points
  - It outputs a weighted sum of the two adjacent sample values
- A 2-tap FIR filter with dynamically varying coefficients
  - Determined by the spline values of the input samples
  - Linear interpolation

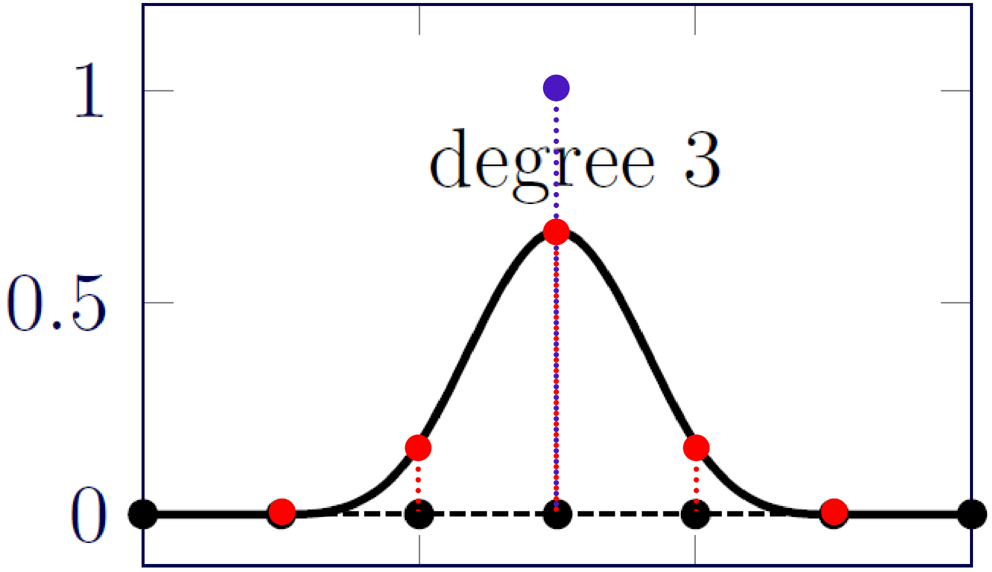


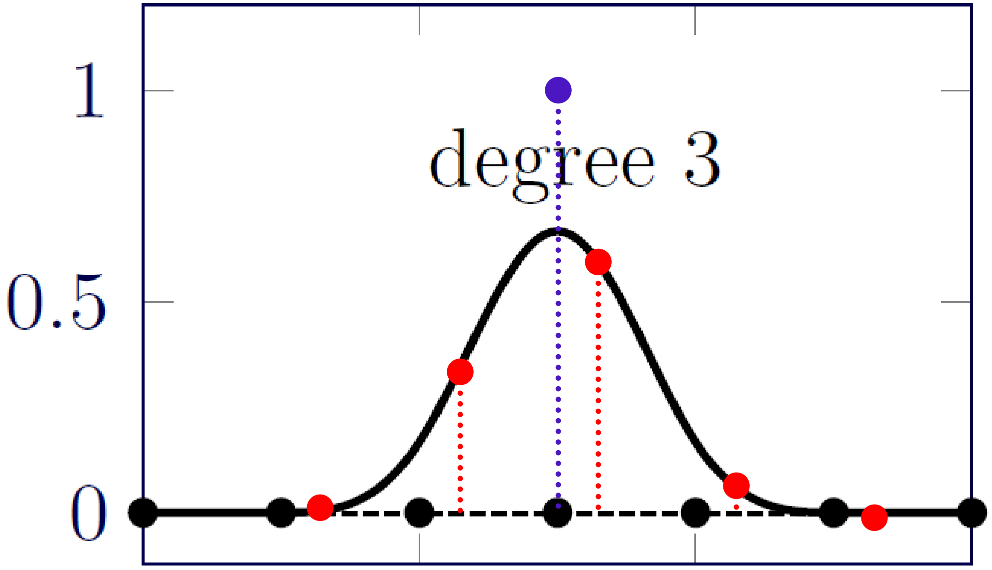



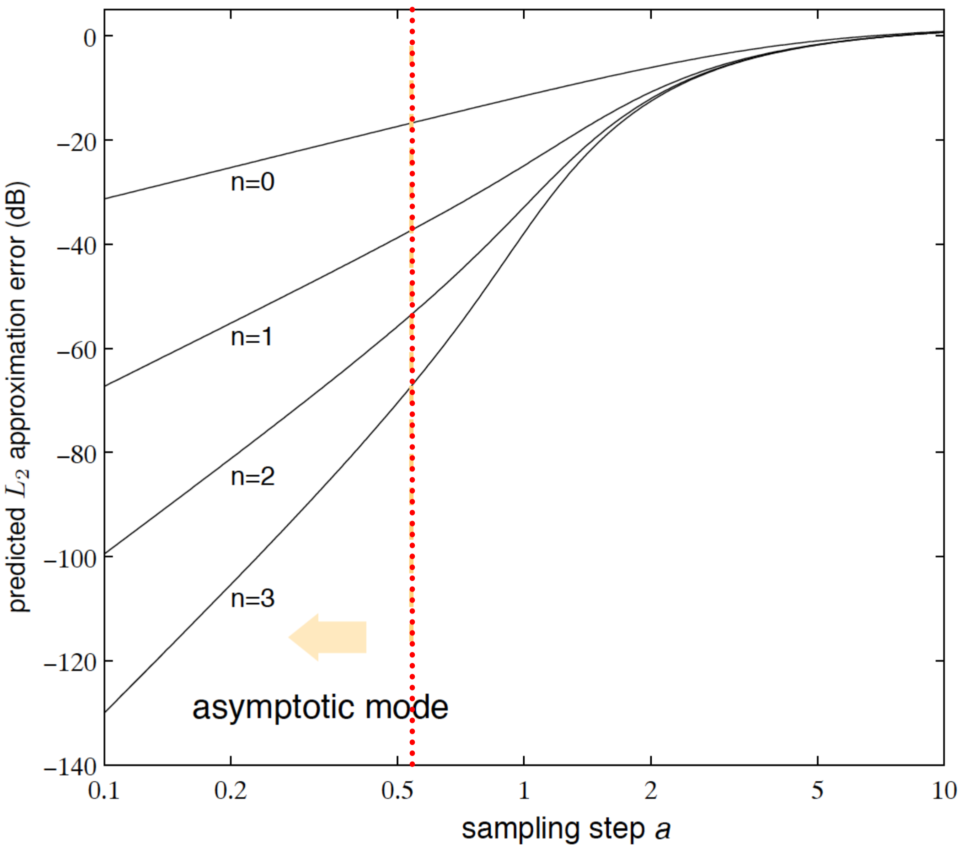
# Filter Based $\beta_3$-Spline Interpolation

- When the $\beta_3$ spline is on a sample point
  - It outputs that sample value
- When the $\beta_3$ spline is between two sample points
  - It outputs a weighted sum of the four adjacent sample values
- A 4-tap FIR filter with dynamically varying coefficients
  - Determined by the spline values of the input samples
  - You can calculate them

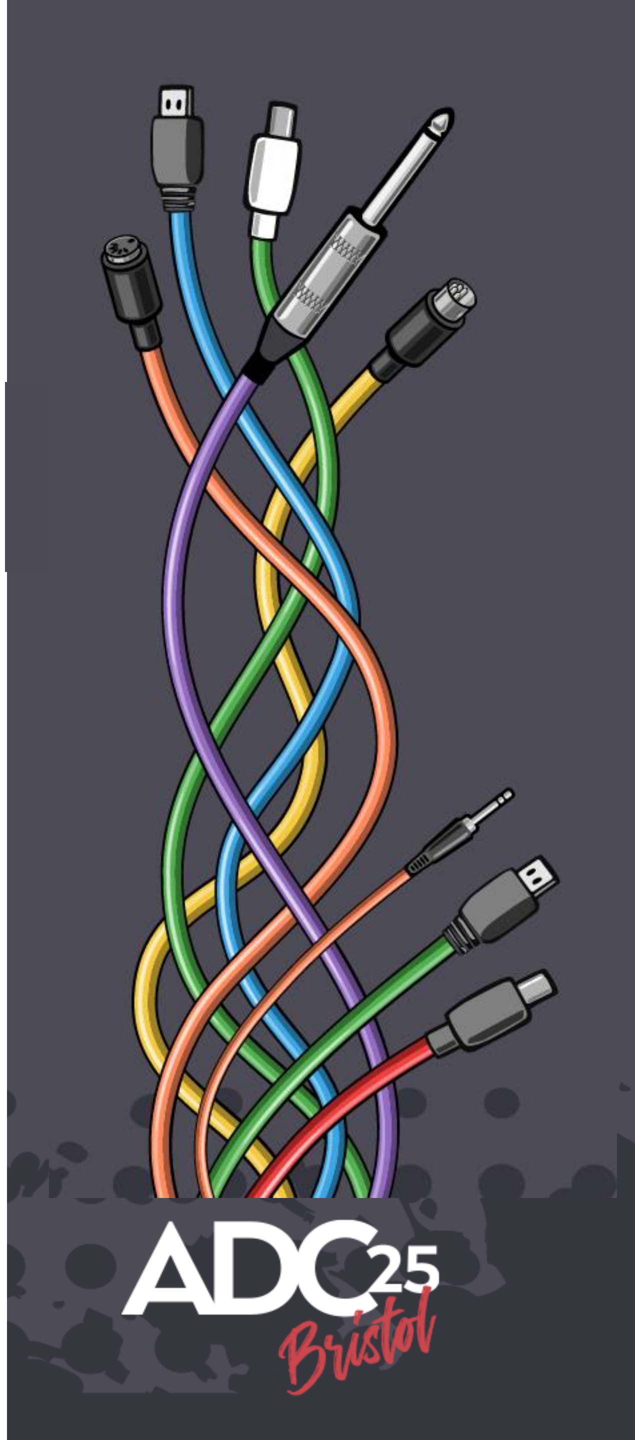


# Using a Filter Instead of Matrix Inversion

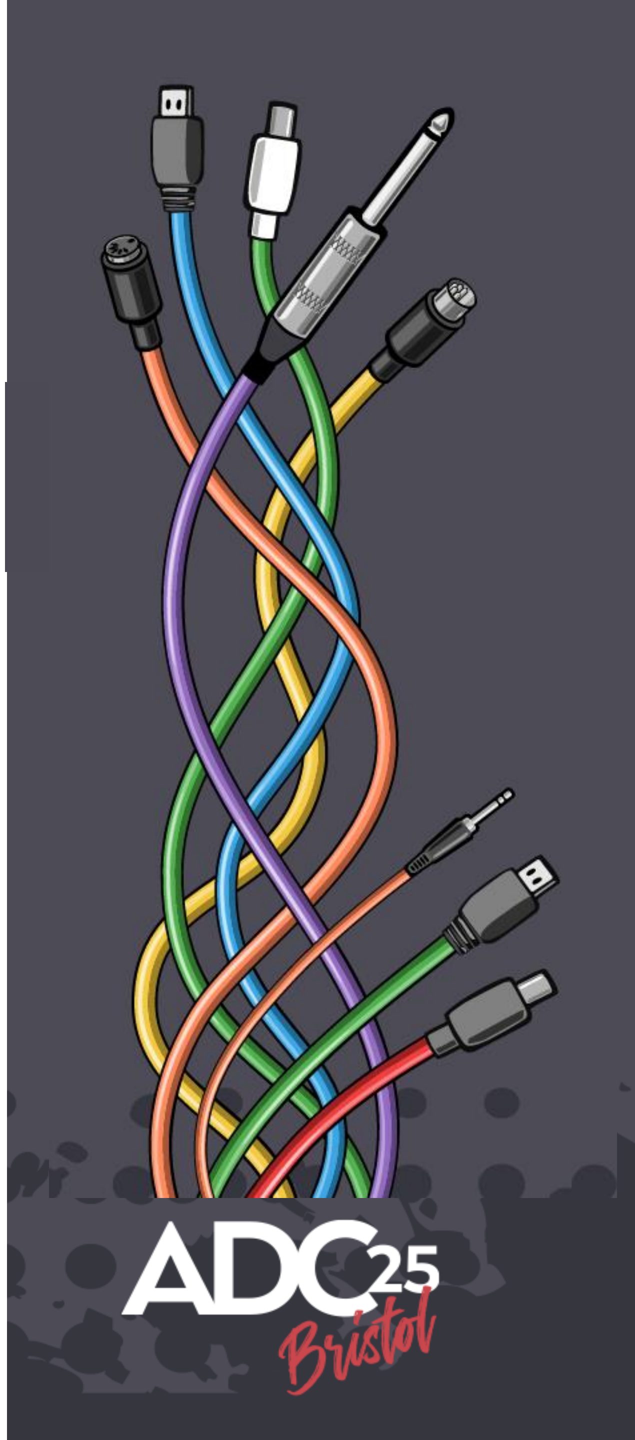



- Oversampling is probably necessary
  - Two times or greater?
  - May be provided already for 96kHz sample rate
- And/or higher order splines

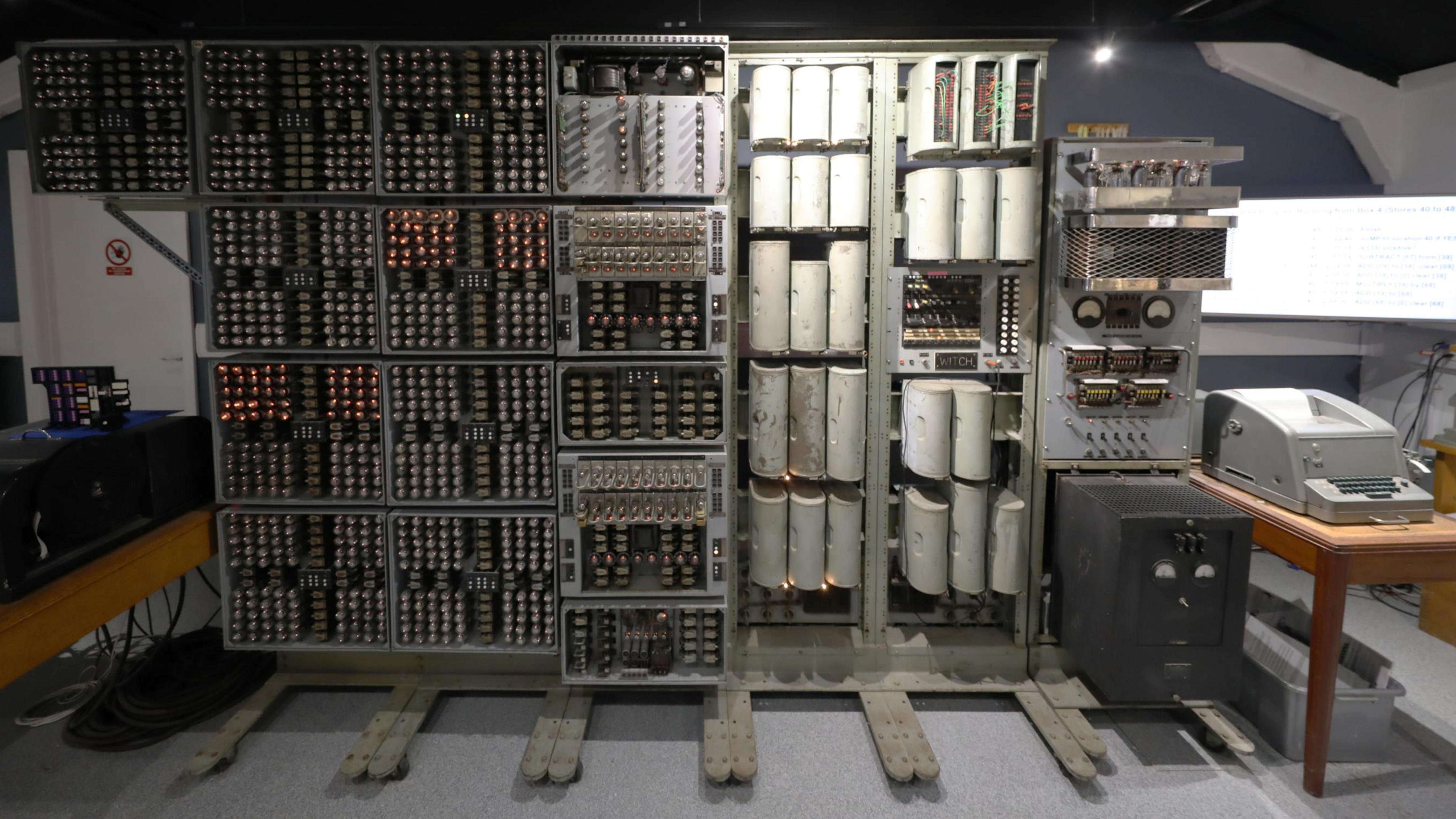


# Conclusion

- The ability to measure and control an audio waveform's level is an important part of many audio devices.
- However, because the audio waveform is sampled
  - The actual level may not be the sample values,
  - But instead, may be an intermediate value between the samples
  - Non-linear operations do not work as expected
  - Dynamic variation in levels are also affected
    - Especially for inter sample peaks
- Various methods have been proposed to ameliorate this
  - Up-sampling
  - Quadratic interpolation
  - β-Spline Interpolation
- The samples are not always what they seem!



# Any Questions?


WITCH