Imperial College London

Lecture 12

Discrete Time Systems

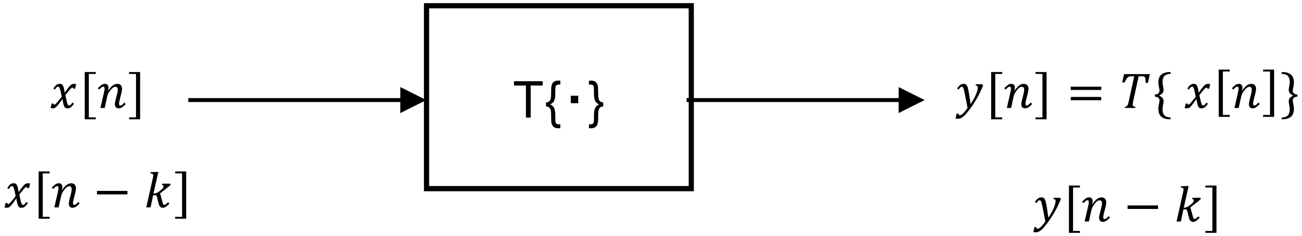
Prof Peter YK Cheung

Dyson School of Design Engineering

URL: www.ee.ic.ac.uk/pcheung/teaching/DE2_EE/
E-mail: p.cheung@imperial.ac.uk

Linear Discrete time systems

- A discrete time system takes in a sequence of discrete values x[n] at the input and produces an output sequence y[n] through some internal operation or transformation T{.}



- The system is LINEAR if it obeys the principle of superposition:

$$y[n] = \; T\{\, a_1 x_1[n] + a_2 x_2[n]\} = a_1 T\{\, x_1[n]\} + a_2 T\{\, x_2[n]\}$$

- The system is shift-invariant if:

$$T\{a_1 x_1[n-k] + a_2 x_2[n-k]\} = y[n-k]$$

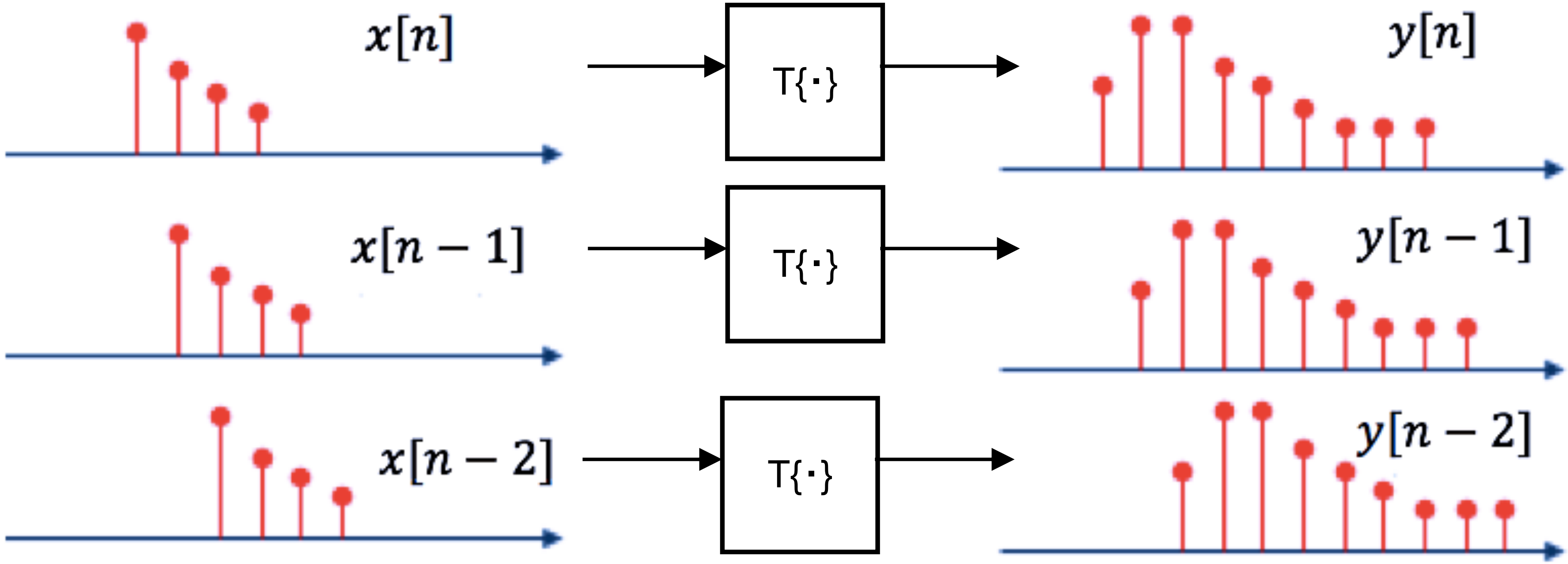
Shift-invariant Discrete time systems

- A system is shift-invariant if delaying the input x[n] by k samples results in the same output y[n], but delayed also by k.

$$x[n] \longrightarrow \boxed{T\{\cdot\}} \longrightarrow y[n] = T\{\,x[n]\}$$

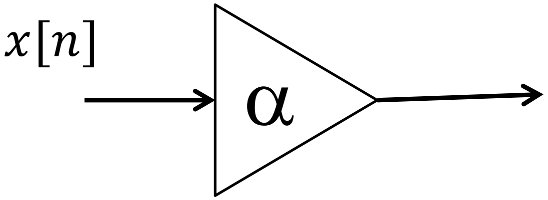
$$x[n-k] \longrightarrow \boxed{T\{\cdot\}} \longrightarrow y[n-k]$$

- In this course, we only consider linear shift-invariant discrete systems.



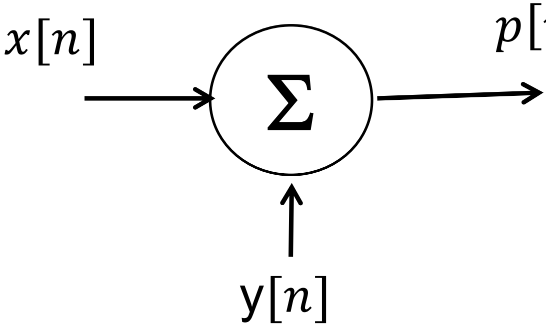
Basic building blocks in a discrete linear system

- Scaling



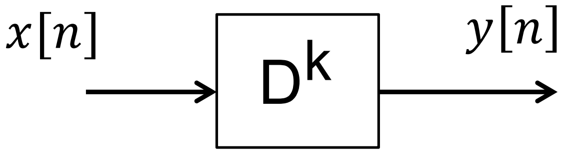
$y[n] = \alpha\, x[n]$

- Adding



$p[n] = x[n] + y[n]$

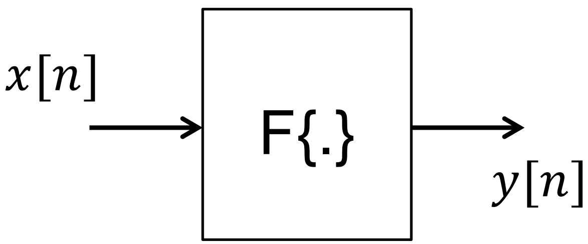
- Delay (i.e. D^k = time shift by k sample periods)

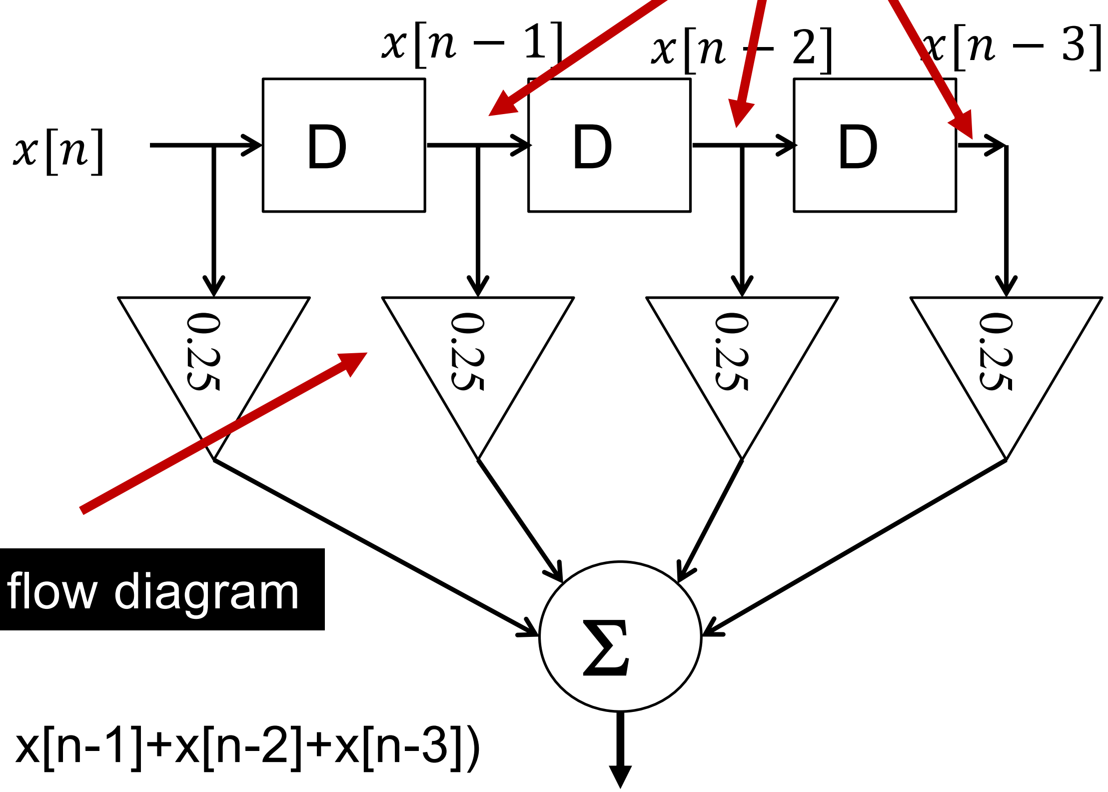


$y[n] = x[n-k]$

Moving average filter

- Consider the following discrete time system:





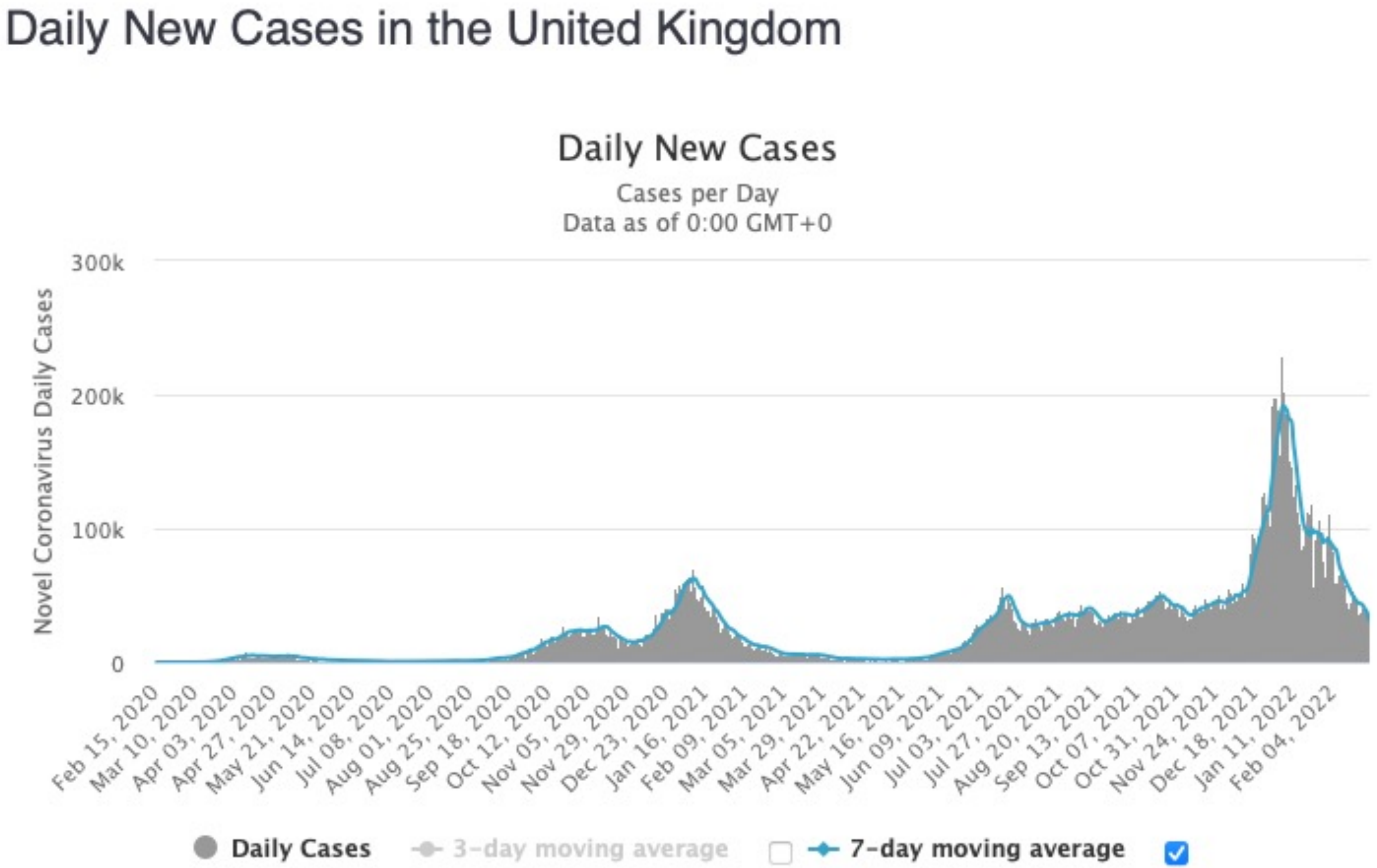
signal flow diagram

difference equation

$y[n]$= 0.25 (x[n] + x[n-1]+x[n-2]+x[n-3])

- This system take the current and the previous 3 input samples, and average them. This is also known as a **moving average filter**.

Example - COVID cases in UK



z-transform and difference equation

- According to Lecture 11 slide 9, if the z-transform of x[n] is X[z]:

$$x[n] \overset{\mathrm{Z}}{\rightarrow} X[z]$$

then,

$$x[n-k] \overset{\mathrm{Z}}{\rightarrow} X[z]\ z^{-k}$$

- In other words, delaying a signal x[n] by k sample period is equivalent to multiplying its z-transform X[z] with z^{-k}.
- We can apply this important property of z-transform (known as the **shift property**) to the difference equation relating the input sequence to the output sequence:

Difference equation ➡ $y[n]= 0.25\ (\mathrm{x}[\mathrm{n}] + \mathrm{x}[\mathrm{n}\text{-}1] + \mathrm{x}[\mathrm{n}\text{-}2] + \mathrm{x}[\mathrm{n}\text{-}3])$

$$Y[z] = 0.25\{X[z] + X[z]z^{-1} + X[z]z^{-2} + X[z]z^{-3}\}$$

z-transform equation ➡ $Y[z] = 0.25(1 + z^{-1} + z^{-2} + z^{-3})X[z]$

- This is the z-domain version of the difference equation in terms of z^{-k}, where k is delay in unit of sample.

Transfer function in the z-domain

- Take the results from the previous slide and re-arrange:

$$y[n] = 0.25\ (x[n] + x[n\text{-}1] + x[n\text{-}2] + x[n\text{-}3])$$

$$Y[z] = 0.25\{X[z] + X[z]z^{-1} + X[z]z^{-2} + X[z]z^{-3}\}$$

$$Y[z] = 0.25(1 + z^{-1} + z^{-2} + z^{-3})X[z]$$

$$H[z] = Y[z]/X[z] = 0.25(1 + z^{-1} + z^{-2} + z^{-3})$$

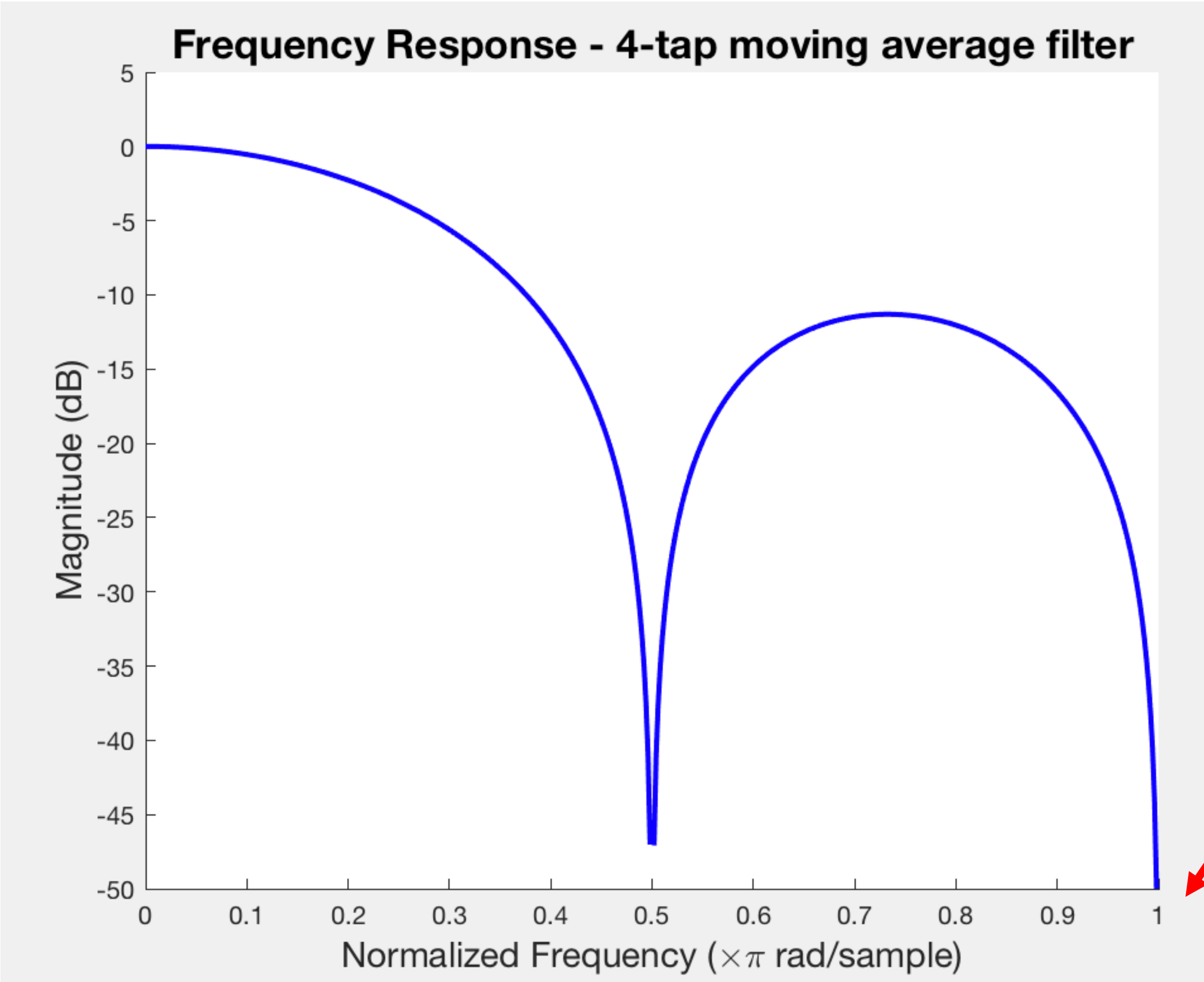
- As in the case of Laplace transform, in the z-domain,

  transfer function H[z] = output Y[z] / input X[z]

- This moving average filter takes the average of the current data sample x[i], and the previous three samples x[i-1], x[i-2] and x[i-3], to produce the output y[i].
- The averaging function has a **smoothing effect** – that is, it performs the function of a **lowpass filter**.

Frequency Response of this filter

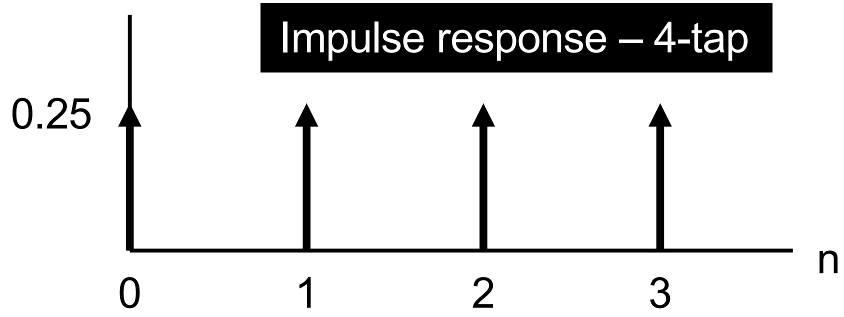
- Here is the frequency response of this moving average filter:



Normalised freq 1 is
Sampling freq Fs / 2

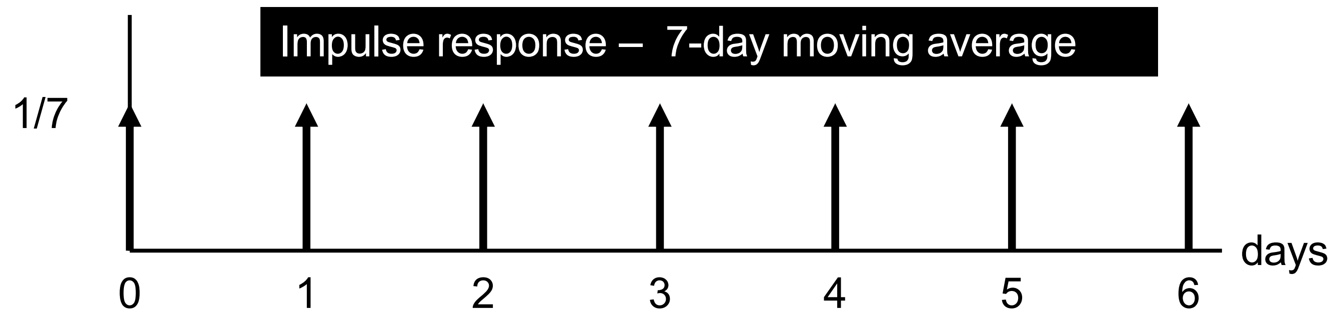
Impulse Response of the moving average filter

- The four-tap moving average filter has an impulse response of:



- The implies that an impulse at n = 0 will have an impact on the output of ¼ on the current sample time, and the three subsequent sample times.

- For the 7-days moving average filter, the impulse response is:



General FIR filters

- Instead of using the same coefficient values in the moving average filter, one could use different coefficients at different delay taps.
- The number of delay taps can be increased to N.
- This will implement a filter function of the form as difference equation:

$$y[n] = b_0x[n] + b_1x[n-1] + b_1x[n-2] + \cdots + b_{N-1}x[n-(N-1)]$$

- In z-domain form:

$$Y[z] = (b_0 + b_1z^{-1} + b_2z^{-2} + b_3z^{-3} + \cdots + b_{N-1}z^{-(N-1)})X[z] = X[z]\sum_{k=0}^{N-1} b_kz^{-k}$$

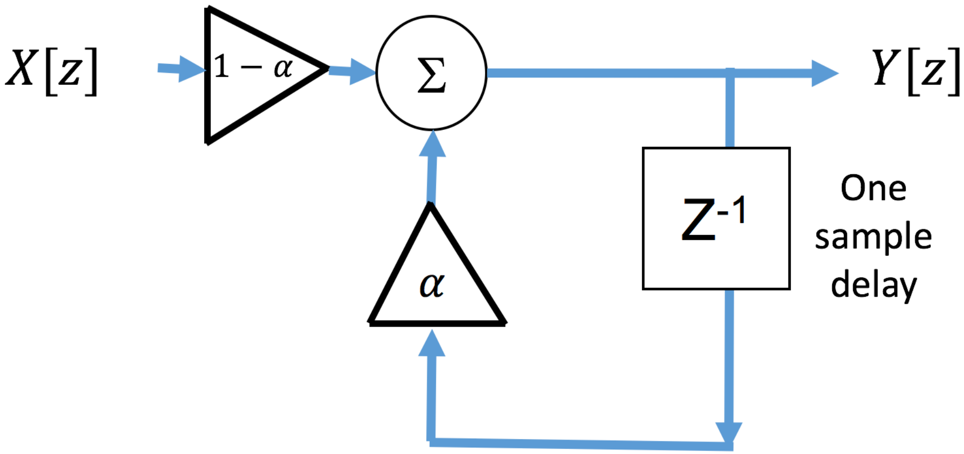
$$H[z] = \frac{Y[z]}{X[z]} = b_0 + b_1z^{-1} + b_2z^{-2} + b_3z^{-3} + \cdots + b_{N-1}z^{-(N-1)} = \sum_{k=0}^{N-1} b_kz^{-k}$$

- By choosing different coefficients b_0, b_1, b_2 …., b_{N-1}, one can implement different types of filters: **lowpass**, **bandpass**, **highpass** etc.
- Such a filter will have N terms in the impulse response, where N is the number of signal taps x[n], … x[n-(N-1)]. Therefore it is also known as a **finite impulse response** filter (FIR) of order **N**.

Recursive Filter

- **FIR filters** derives the current output from **current** and **previous** inputs
- Such a filter does not make use of previous outputs – that is, it does not rely on past information
- **Recursive filter** or **IIR filter** is different – it derives the current output from **both input and previous output** samples.
- Name of IIR comes the fact that its impulse response is in finite.
- Here is one of the simplest recursive filter:

$$y[n] = \alpha y[n-1] + (1-\alpha)x[n]$$



$$Y[z] = \alpha\, Y[z]\, z^{-1} + (1-\alpha)X[z]$$

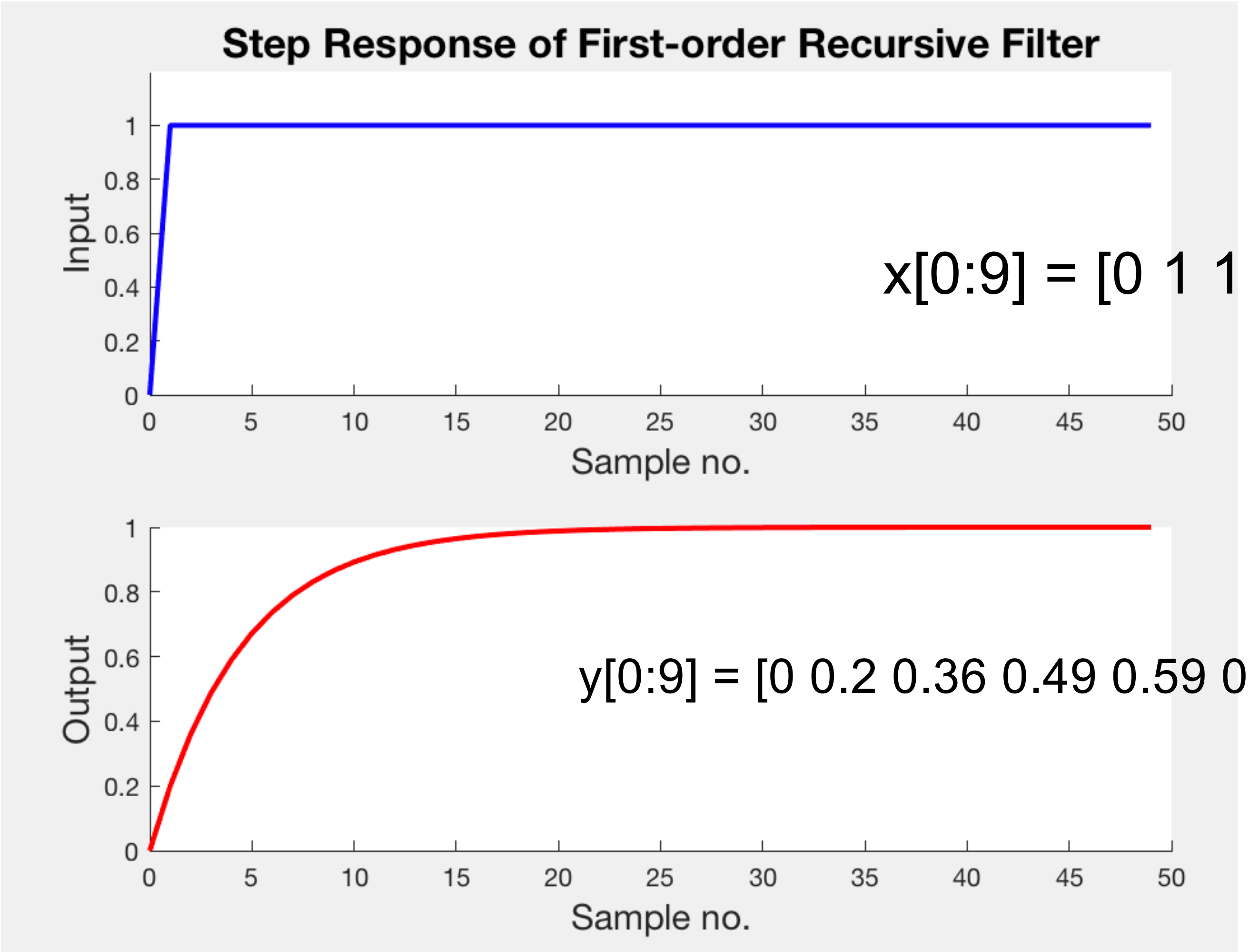
$$\Rightarrow Y[z] - \alpha\, Y[z]\, z^{-1} = (1-\alpha)X[z]$$

$$\Rightarrow (1-\alpha\, z^{-1})Y[z] = (1-\alpha)X[z]$$

$$\Rightarrow H[z] = \frac{Y[z]}{X[z]} = \frac{(1-\alpha)}{(1-\alpha\, z^{-1})}$$

Step response of Recursive Filter

- Let us consider the response of this filter to unit step function at the input



x[0:9] = [0 1 1 1 1 1 1 1 1 1]

y[0:9] = [0 0.2 0.36 0.49 0.59 0.67 0.74 0.79 0.83 0.87]

Three Big Ideas

1. A discrete-time system can be represented in three ways:
   - **Block diagram** (see slide 5)
   - **Difference equation**, e.g. $y[n]$= 0.25 (x[n] + x[n-1]+x[n-2]+x[n-3])
   - **Impulse response** in the z-domain, e.g. $H[z] = \frac{1}{4}(1 + z^{-1} + z^{-2} + z^{-3})$
2. A **Finite Impulse Respo**nse filter (FIR), as the name implies, is one where the effect of input signal lasts for fixed number of sampling periods before reaching zero. However **Infinite Impulse Respnse** (IIR) filter in theory has an infinite impulse responses the lasts forever, but may decay towards zero.
3. An **Infinite Impulse Respnse** (IIR) filter is much more efficient to implement than FIR filter. For the same filtering effect, IIR filter has much lower number of multiplier and add operations than FIR filter. However, it badly designed IIR filter may be unstable.