EEE6440
Advanced Signal Processing (ASP)

- Digital Filters II:
  - Time/Frequency domain parameters
  - What makes a good filter?
  - Realising other filters from a LPF
  - Moving average filter

- MATLAB
  - Commands: fir1, fir2, remez, firls

Dr Charith Abhayaratne

Tel: 25893
Email: c.abhayaratne@sheffield.ac.uk
Office: F176

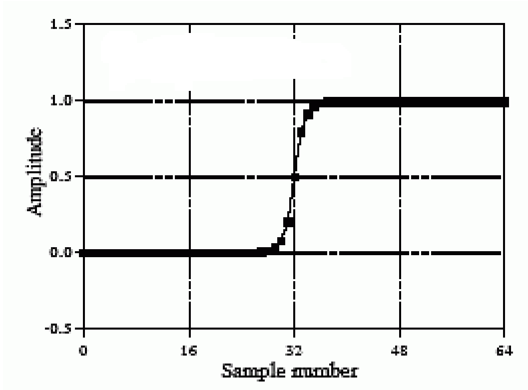
Information in signals

- There are many ways that information can be contained in a signal.
  - information represented in the time domain,
  - information represented in the frequency domain
- Information represented in the **time domain** describes when something occurs and what the amplitude of the occurrence is.
- The **step response** describes how information represented in the **time domain** is being modified by the system

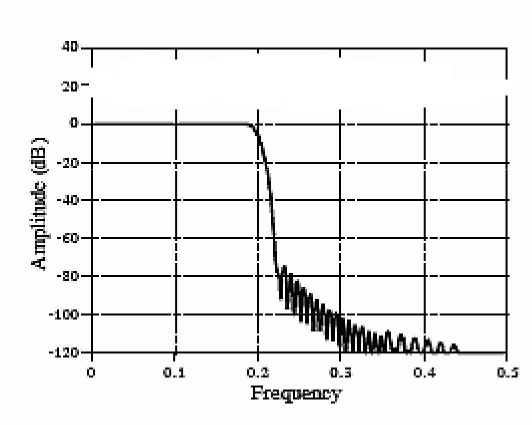
- The **frequency domain** information represent information with regards to periodic nature of signal components.
- A single sample, in itself, contains no information about the periodic motion. The information is contained in the relationship between many points in the signal.
- The **frequency response** shows how information represented in the **frequency domain** is being changed.

Filter parameters

- Step response
  - Fast step response
  - No overshoot
  - Linear phase



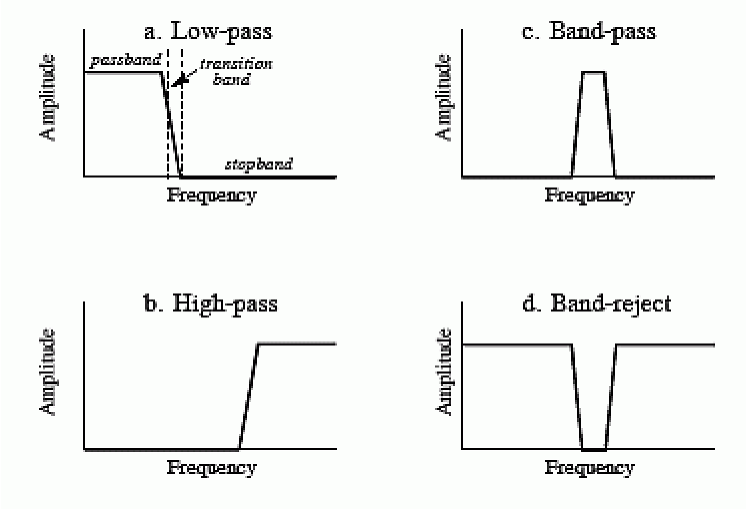
- Frequency response
  - Fast roll-off
  - Flat passband
  - Good stopband attenuation



- Why step response and not the impulse response is considered?
- What about phase related parameters of the frequency response?

Filter Classification

- Four main types
  - Low pass
  - High pass
  - Band pass
  - Band reject



- How to design a high pass filter if we know the impulse response of the low pass FIR filter h(n)?

- Show how to use cascade and/or parallel LPF and/or HPF to realise band-pass and band-reject filters.

- Filters can be summarized by
  - Their use
    - 1) In time domain
      - E.g. ?
    - 2) In frequency domain
      - E.g. ?
    - 3) Custom
      - E.g. ?
  - Their implementation
    - A) Convolution
    - B) Recursion

- Examples for 1A, 1B, 2A, 2B, 3A, 3B ?

The Moving average filter (MAF)

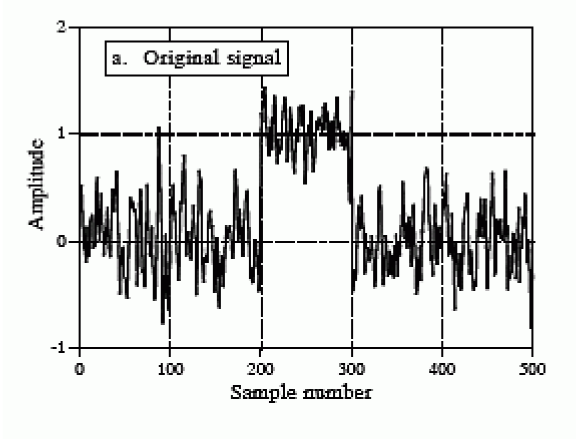
- The moving average filter operates by averaging a number of points from the input signal to produce each point in the output signal.

$$y(i) = \frac{1}{M} \sum_{k=0}^{M-1} x(i+k)$$

- Example: Consider a 5-point MAF, i.e., M=5
  - What is the impulse response?
  - What is the step response?
  - Write a pseudo-code to implement it
  - What is the frequency response?

- The main aim of the MAF is to remove the noise while keeping sharp transitions intact

- Noise reduction performance
  - Consider the signal below



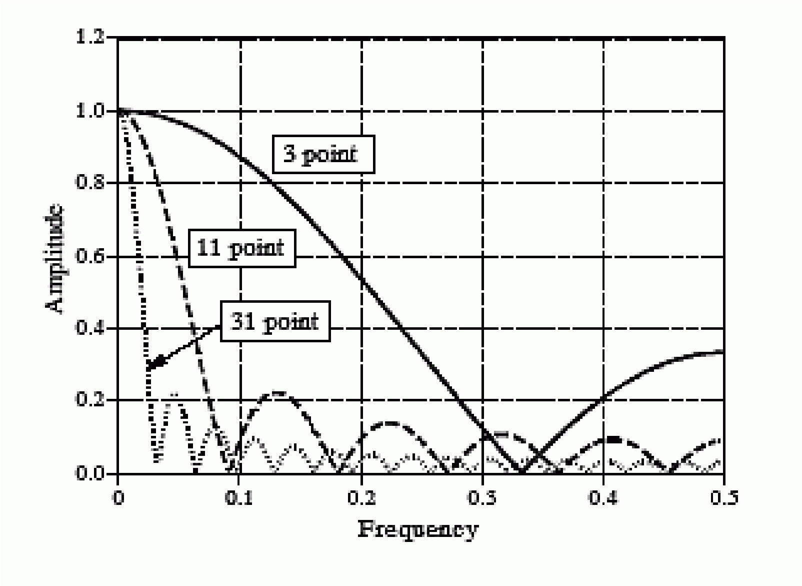
MATLAB commands:
X=[zeros(1,200) ones(1,100) zeros(1,200)];
Y= X+ 0.2*randn(1,500); % original signal

  - What is the output (recovered X) when the following three MAFs are used on Y.
    - M=5
    - M=11
    - M=25
  - Explain your observations using the step and frequency responses of the above filters.

- Frequency response of MAF
  - Mathematically described by the Fourier transform of the rectangular pulse.
  - How to write the rectangular pulse ?
  - Frequency response?

$$H[f] = \frac{\sin(\pi f M)}{M \sin(\pi f)}$$

  - An alternative way
    - zero pad the impulse response to make it a length N filter, where N is a power of 2
    - Take the FFT
    - 0 to N/2 points represent the Frequency response from 0 to 0.5 of the normalised frequency.



- The roll-off is very slow and the stopband attenuation is ghastly. Clearly, the moving average filter cannot separate one band of frequencies from another. (Good performance in the time domain results in poor performance in the frequency domain, and vice versa)

- the moving average is an exceptionally good *smoothing filter* (the action in the time domain), but an exceptionally bad *low-pass filter* (the action in the frequency domain).

- Multiple pass MAF

  - Consider M=5 MAF and involve passing the input signal through a moving average filter two or more times.
  - Find out the
    - Filter kernal (the impulse response)
    - Step response
    - Frequency response

    For 1 pass, 2-pass and 3-pass MAF.
  - How is the noise removal performance for the (example in slide 12)
    - For 2-pass MAF
    - For 3-pass MAF

- Multiple pass MAF

  - Consider the output after the first stage

$$y_1 = h * x$$

  - Second stage

$$y_2 = h * y_1$$

$$y_2 = h * h * x$$

$$h_2 = h * h$$

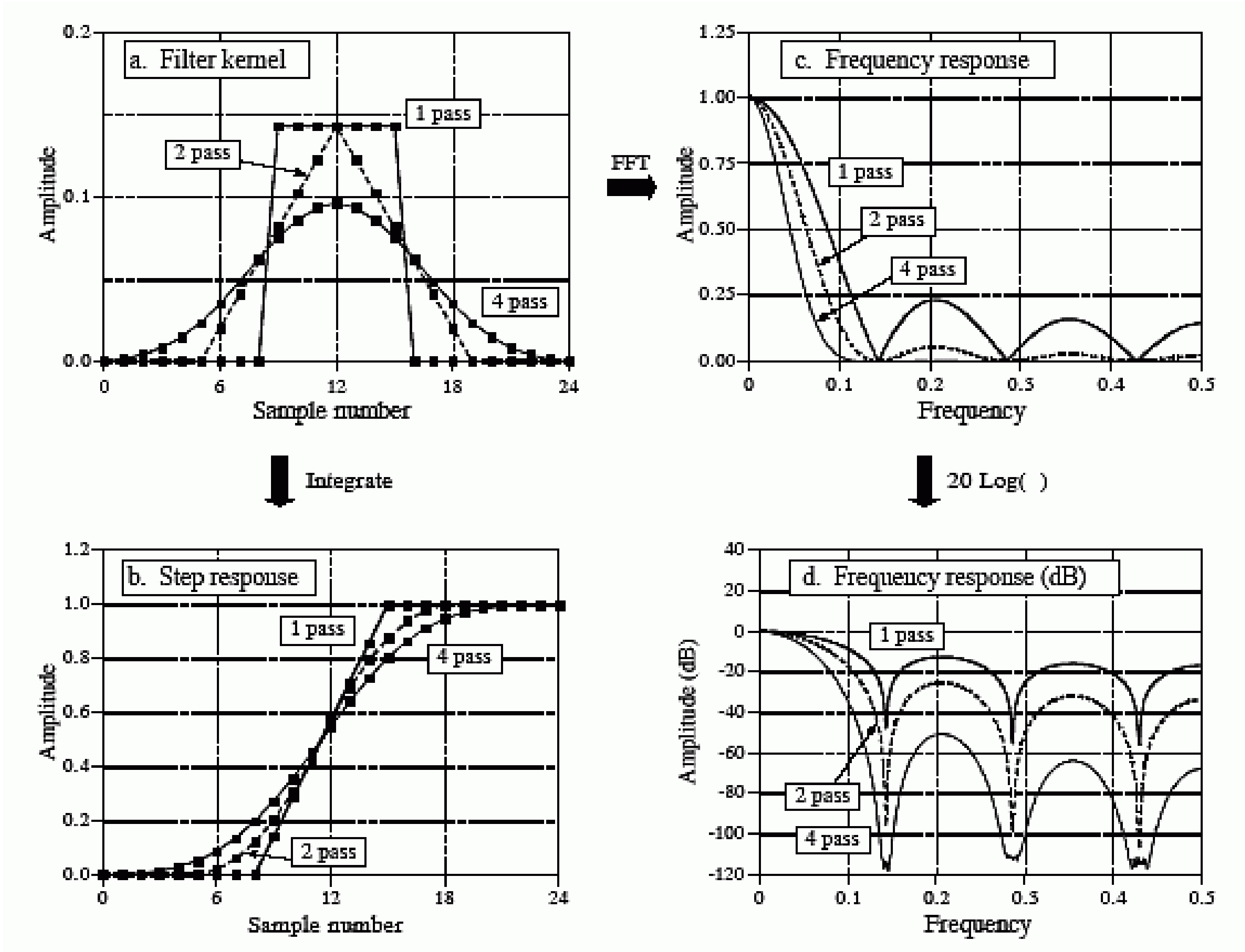
  - Third stage

$$y_3 = h * y_2$$

$$y_3 = h * h_2 * x$$

$$h_3 = h * h_2$$

- Multiple pass MAF



- Multiple pass MAF

  - Two passes are equivalent to using a *triangular* filter kernel (a rectangular filter kernel convolved with itself).
  - After four or more passes, the equivalent filter kernel looks like a *Gaussian*.
  - As shown in (b), multiple passes produce an "s" shaped step response, as compared to the straight line of the single pass.
  - The frequency responses in (c) and (d) are given by the equation in slide 14 *multiplied* by itself for each pass. That is, each time domain convolution results in a multiplication of the frequency spectra.
  - Why are the resulting filters from multiple passes better than the MAF itself?

- Recursive implementation of MAF

  - Consider a MAF with M=5
  - y[n] is computed as

$$y[n] = (x[n-2]+x[n-1]+x[n]+x[n+1]+x[n+2])/5$$

  - y[n+1] is computed as

$$y[n+1] = (x[n-1]+x[n]+x[n+1]+x[n+2]+x[n+3])/5$$

In other words,

$$y[n+1] = y[n]+(x[n+3]-x[n-2])/5$$

  - For any M, y[i] is computed as

$$y[i] = y[i-1]+(x[i+p]-x[i-q])/M$$

where $p=(M-1)/2$

$q=p+1$

- Recursive implementation of MAF
  - The equation in the previous slide uses two sources of data to calculate each point in the output:
    - points from the input *and*
    - previously calculated points from the output.
  - This is called a **recursive** equation, meaning that the result of one calculation is used in *future* calculations.
  - Be aware that the moving average recursive filter is very different from typical recursive filters.
  - Is this an IIR or a FIR?
  - This algorithm is faster than other digital filters for several reasons.
    - there are only two computations per point, regardless of the length of the filter kernel.
    - addition and subtraction are the only math operations needed (if division by M is omitted at the individual step and performed at the end) --- Integer representation.

- Custom filters
  - Most filters have one of the four standard frequency responses: low-pass, high-pass, band-pass or band-reject.
  - But most of the time, we have to design digital filters with an *arbitrary* frequency response, tailored to the needs of our particular application.
  - Two important uses of custom filters:
    - *Deconvolution* - a way of restoring signals that have undergone an unwanted convolution, and
    - *Optimal filtering* - the problem of separating signals with overlapping frequency spectra. (lectures by WL)

- Deconvolution
  - The detected signal is the desired signal convolved with an unwanted kernel.
  - How do we recover the desired signal back?
  - If the exact frequency response of the desired signal not known how can we get the desired signal back? (blind deconvolution - from WL lectures)