Filtering

awaiting

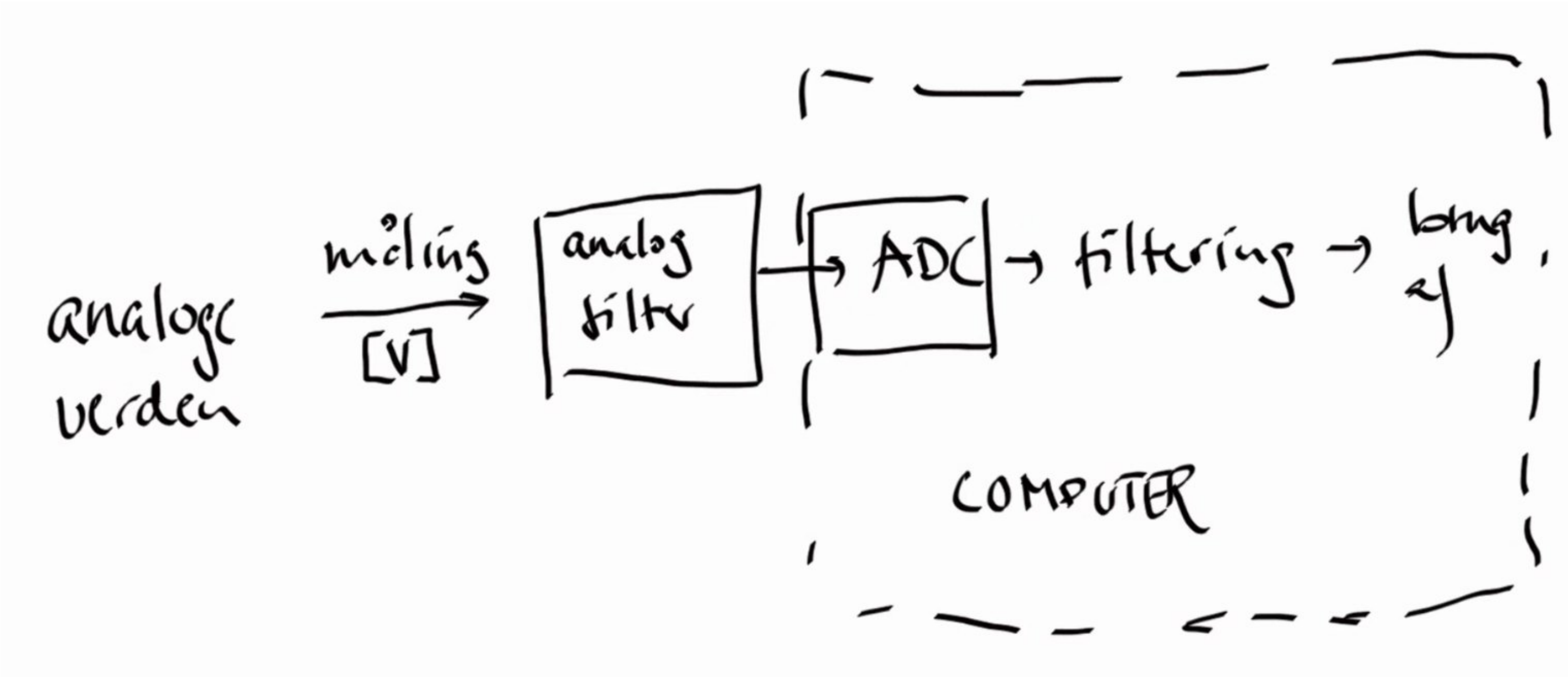
Pre comments

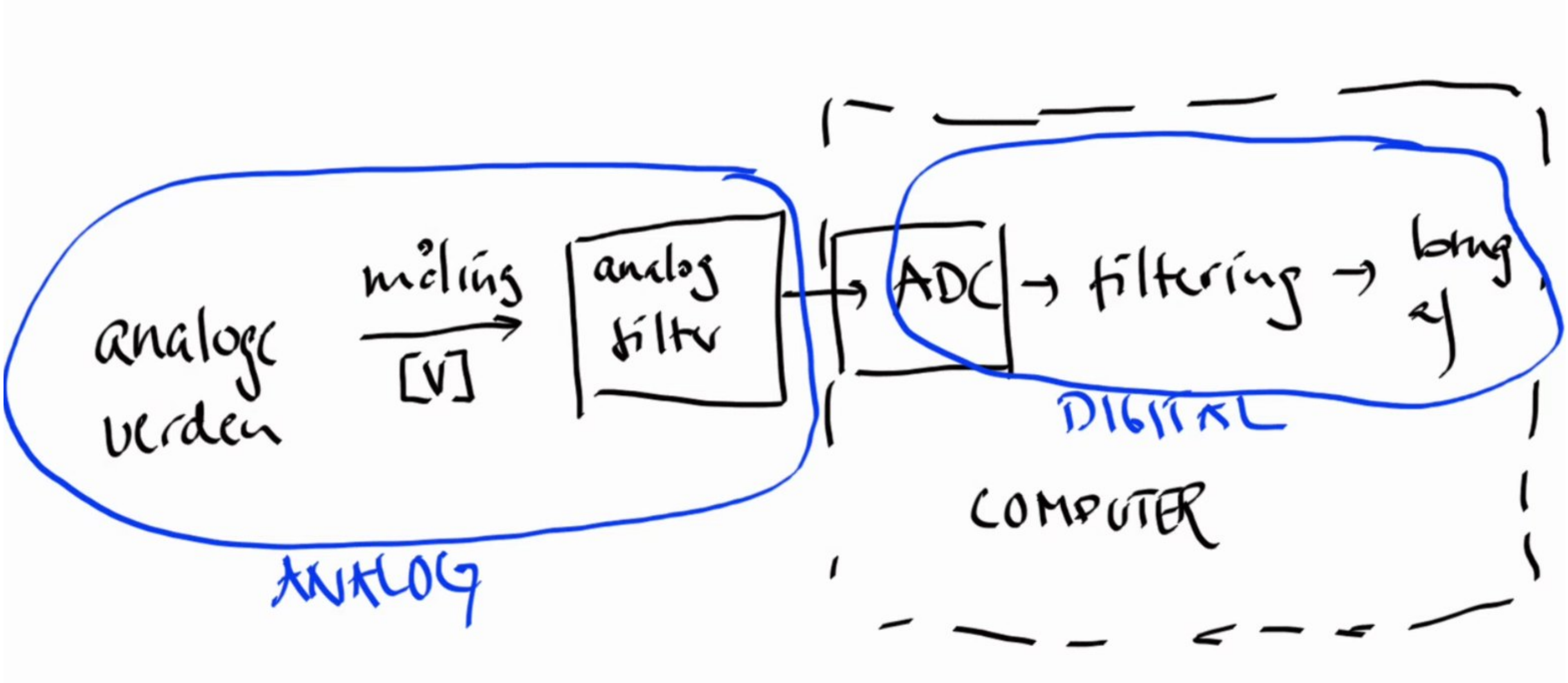
Filtrering og tilpasning er stort - meget stort emneområde

Formålet idag er at få en indsigt i området og gøre jer operationelle

oooOooo

fra jord til bord




analoge verden
måling
[V]
analog filter
ADC
filtering
brug af
DIGITAL
COMPUTER
ANALOG

det store problem

Digitalisering (sampling) er underlagt især en naturlov

Shanning-Nyquists sampling teorem

Det analog signal må **ikke** indeholde frekvenser større end det halve af samplingsfrekvensen

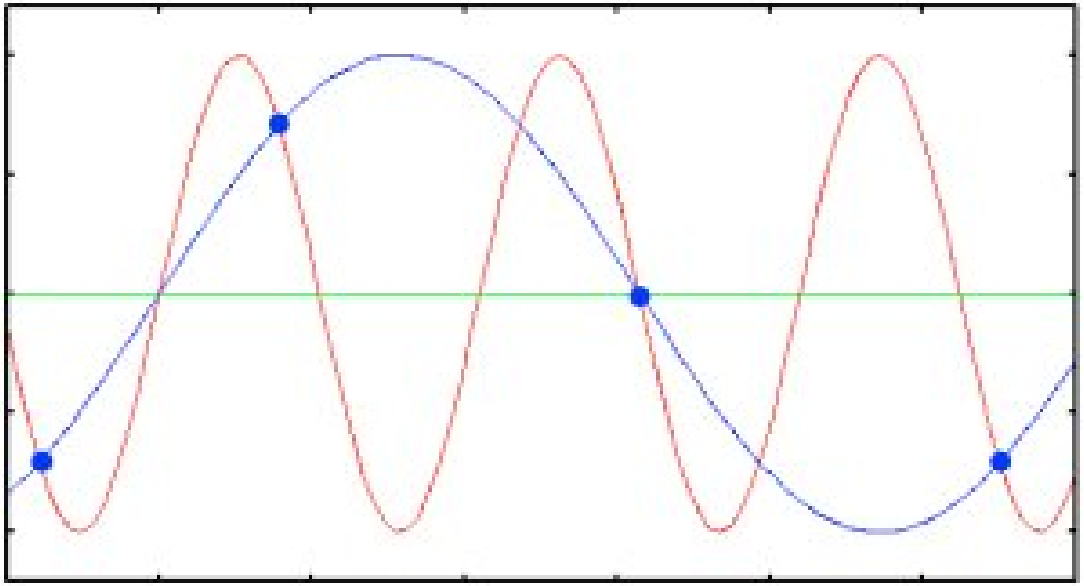
Er det tilfældet

har vi aliasing problem

- Aliasing et eksempel !!

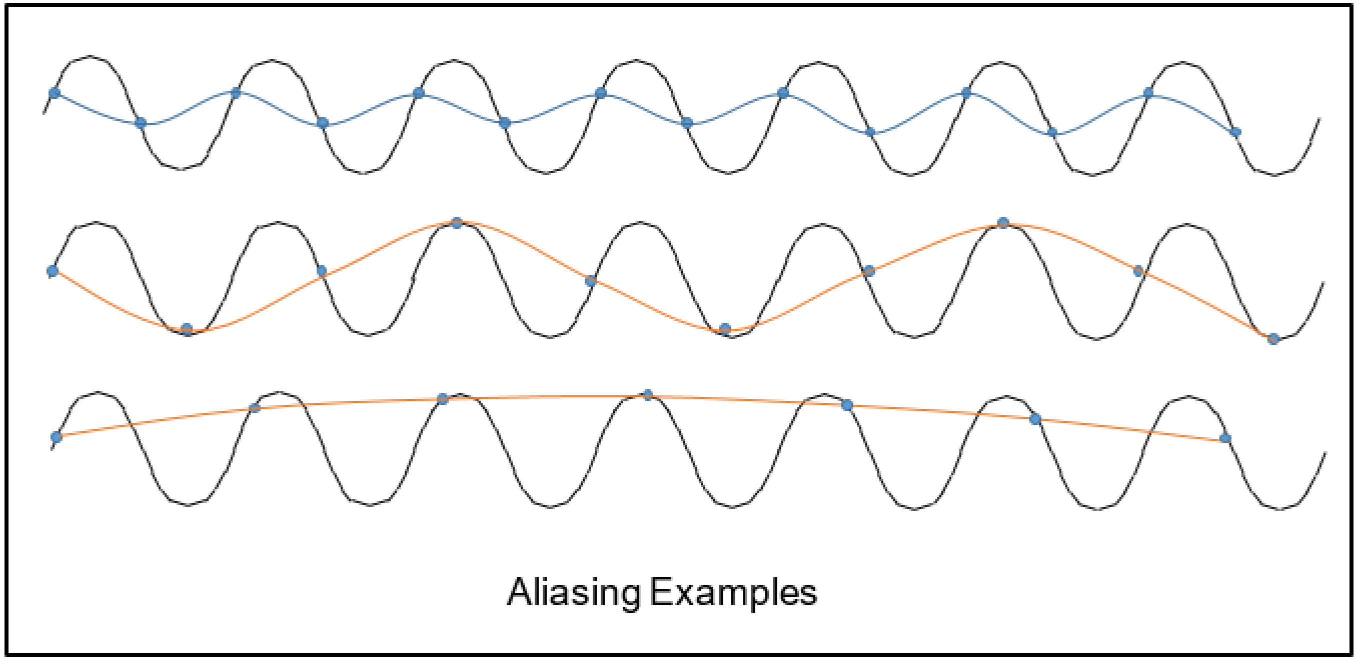
- rød signal - det (rigtige) analoge signal
- blå signal - samplet signal i computer

- HMMMM



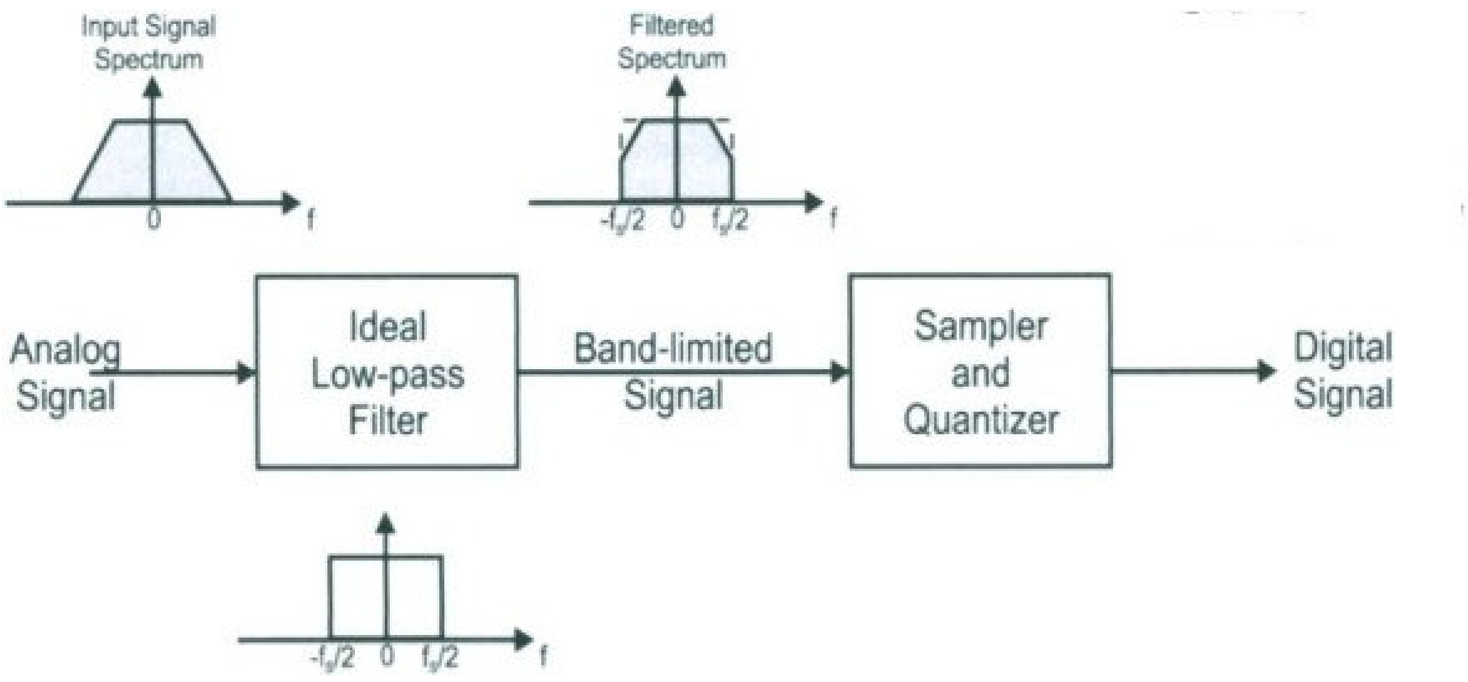
dataforth.com

- øverst ingen alias, men grænsetilfælde
- to andre er hårdt ramt af alias



!

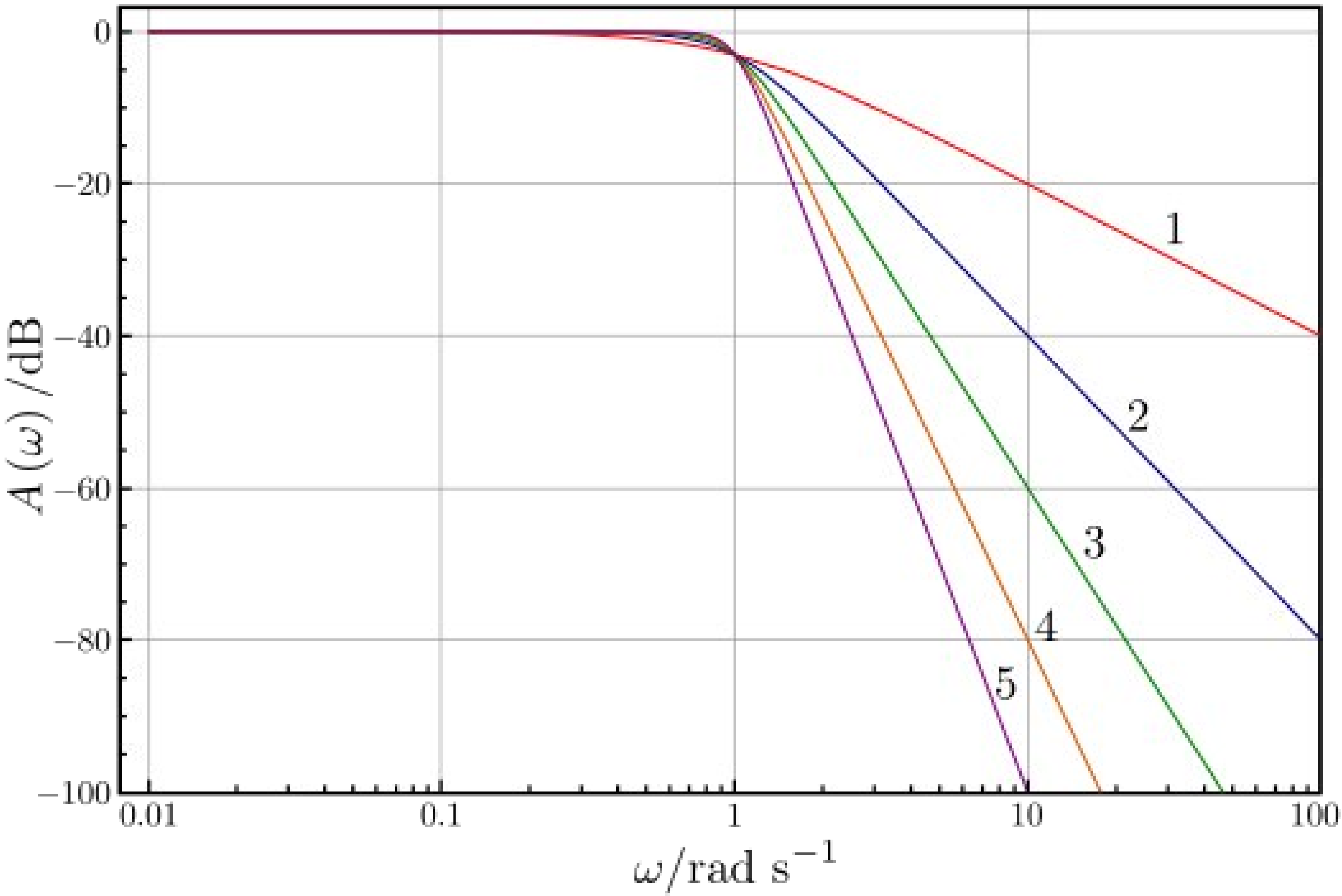
- The Nyquist-Shannon sampling theorem states that
- to restore a signal, a sufficient sample rate must be greater than twice the highest frequency of the signal being sampled.
- er der mere at sige ?
- udover vi skal have begrænset det analoge signal
- As the figure below indicate the lowpass filter (the antialiasing filter) should/should be of very high order - but this is in most /many cases unrealistic.



Butterworth familiy lowpass filters

- flat in pass band
- -20dB ~ 0.1x level
- -40dB ~ 0,01x
- -80dB ~ 0,0001x

first order	6 dB /octave	20 dB/decade
second order	12 dB /octave	40 dB/decade
third order	18 dB /octave	60 dB/decade
fourth order	24 dB /octave	80 dB/decade



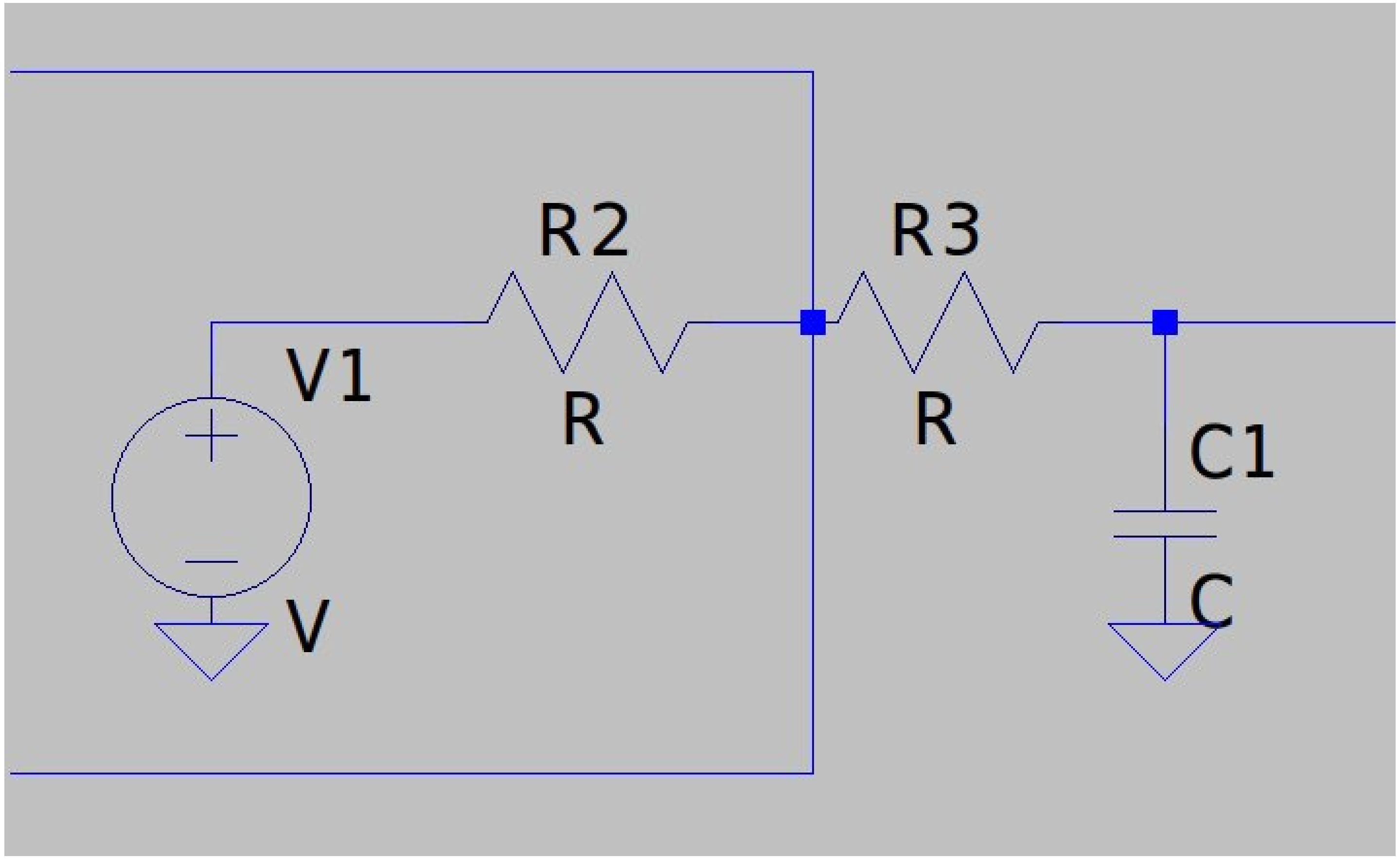
Kvantisering

- 10 bit ADC ~ 1000(10^3) niveauer delta dB : 60 dB fra max til min
- 12 bit ADC ~ 10^5 :   100dB fra max til min

- med nominel udstyring på -10dB er der 10dB mindre afstand til støjgulvet

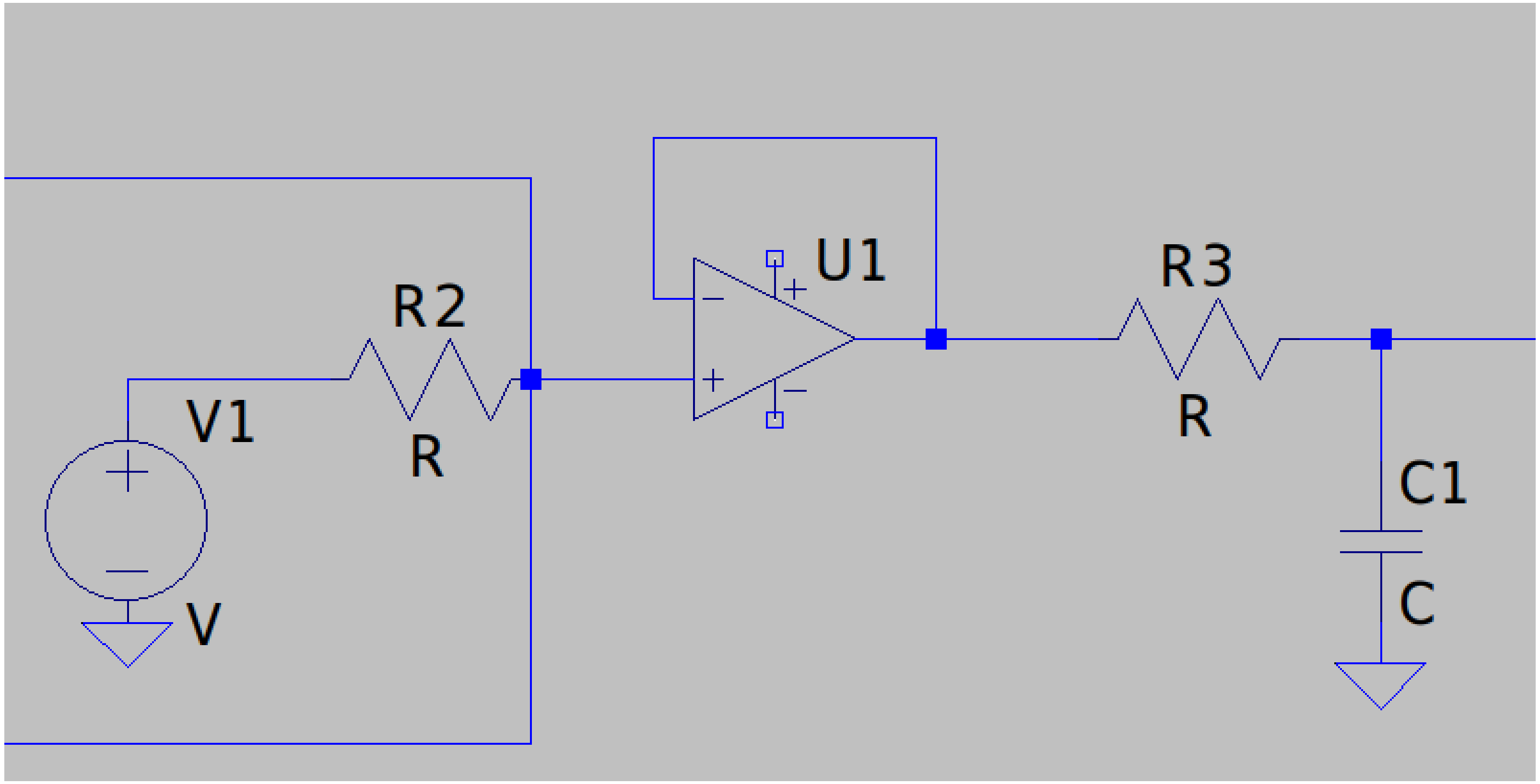
- Så i best case skal signal dæmpes til under kvantiseringsniveau

Impl



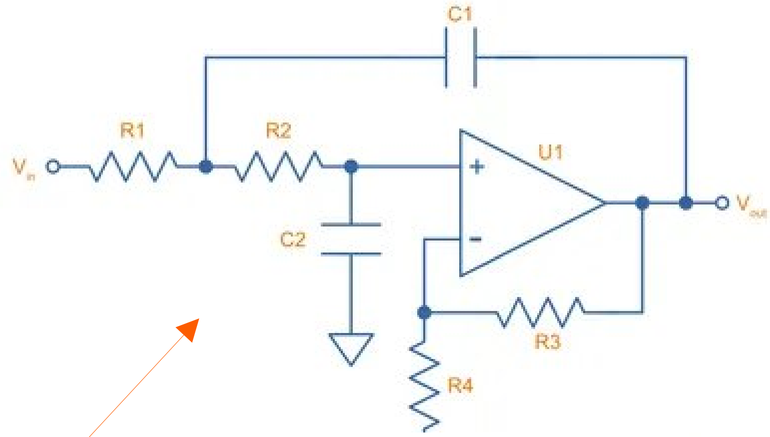
Impdeansovervejelser

- Hvis R_{output} fra forrige trin er sammenlignelig med R3 buffering skal overvejes



højere ordens

- 2^{nd} ordens
- kan kaskadekobles til endnu højere orden
- jeg vil nook aldrig gå højere end 3 orden
- istedet vil jeg sample hurtigere



n	Factors of Butterworth Polynomials $B_n(s)$
1	$(s+1)$
2	$(s^2+1.414214s+1)$
3	$(s+1)(s^2+s+1)$
4	$(s^2+0.765367s+1)(s^2+1.847759s+1)$
5	$(s+1)(s^2+0.618034s+1)(s^2+1.618034s+1)$
6	$(s^2+0.517638s+1)(s^2+1.414214s+1)(s^2+1.931852s+1)$
7	$(s+1)(s^2+0.445042s+1)(s^2+1.246980s+1)(s^2+1.801938s+1)$
8	$(s^2+0.390181s+1)(s^2+1.111140s+1)(s^2+1.662939s+1)(s^2+1.961571s+1)$
9	$(s+1)(s^2+0.347296s+1)(s^2+s+1)(s^2+1.532089s+1)(s^2+1.879385s+1)$
10	$(s^2+0.312869s+1)(s^2+0.907981s+1)(s^2+1.414214s+1)(s^2+1.782013s+1)(s^2+1.975377s+1)$

anti aliasing

- NB: det er (meget) ofte noget man glemmer at kigge på

- en måde

- oscilloskop på og måle på sensor
- frekvensplot vha oscilloskopet

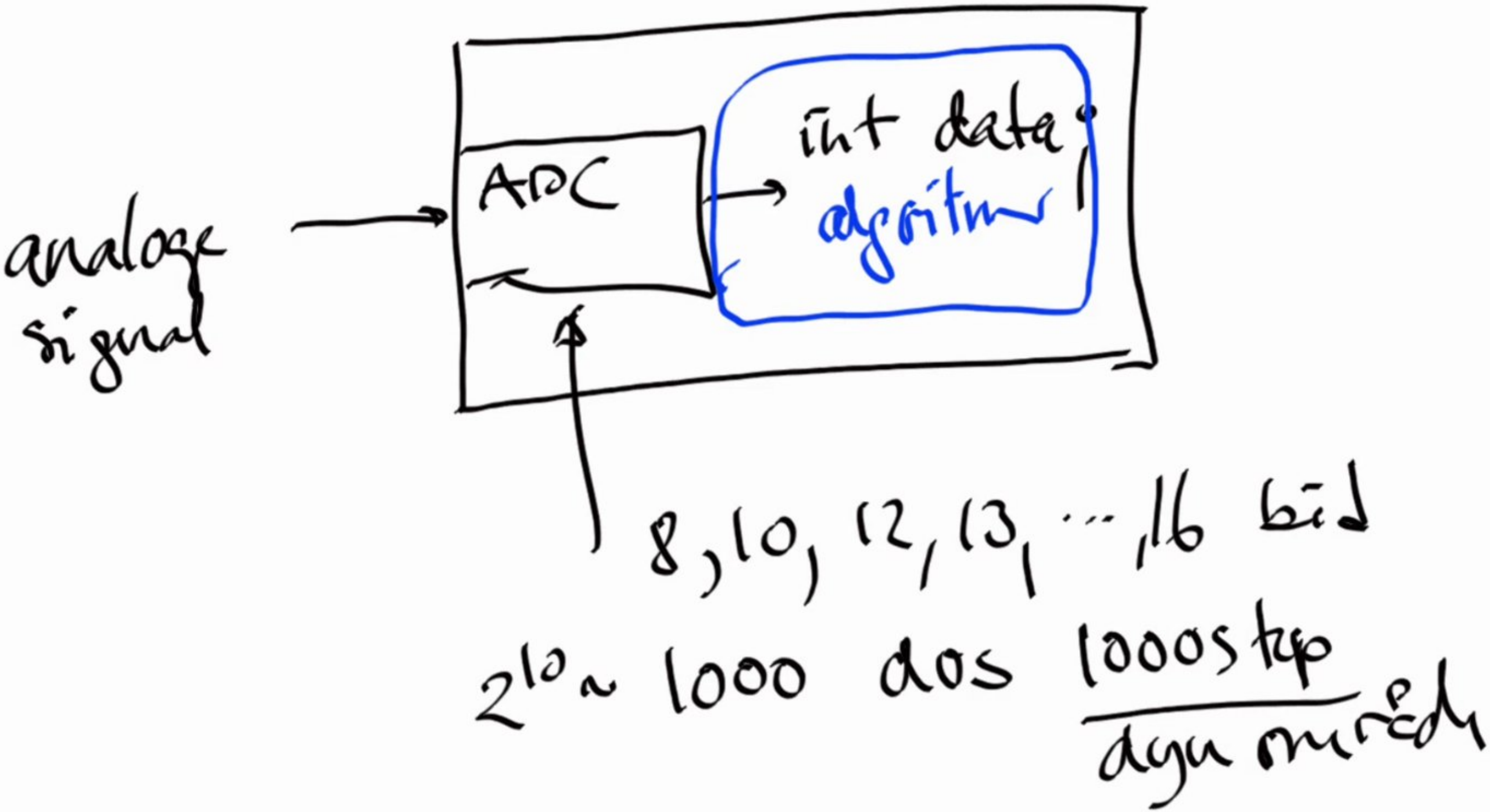
- vurdere om der er aliasing
  - du kan se om det er frekvenser over den ønskede sampligns frekvens (gang med 0.5)
- Skal du bruge de høje frekvenser
- så skal samplingsfrekvens op.

- https://www.youtube.com/watch?v=ACWa_mMhSJs

Støj

- Støj i på signaler fra den virkelige verden hører sig lidt til
- 2 grupper
- støj indenfor det frekvensområde man vil arbejde - ikke trivielt at fjerne
- støj udenfor det ønskede arbejdsområde - dagens tekst
- afgrænsning til støj i frekvens over vores arbejdsområde.

herfra er vi i den digitale verden.



Simple Moving Average filter (SMA)

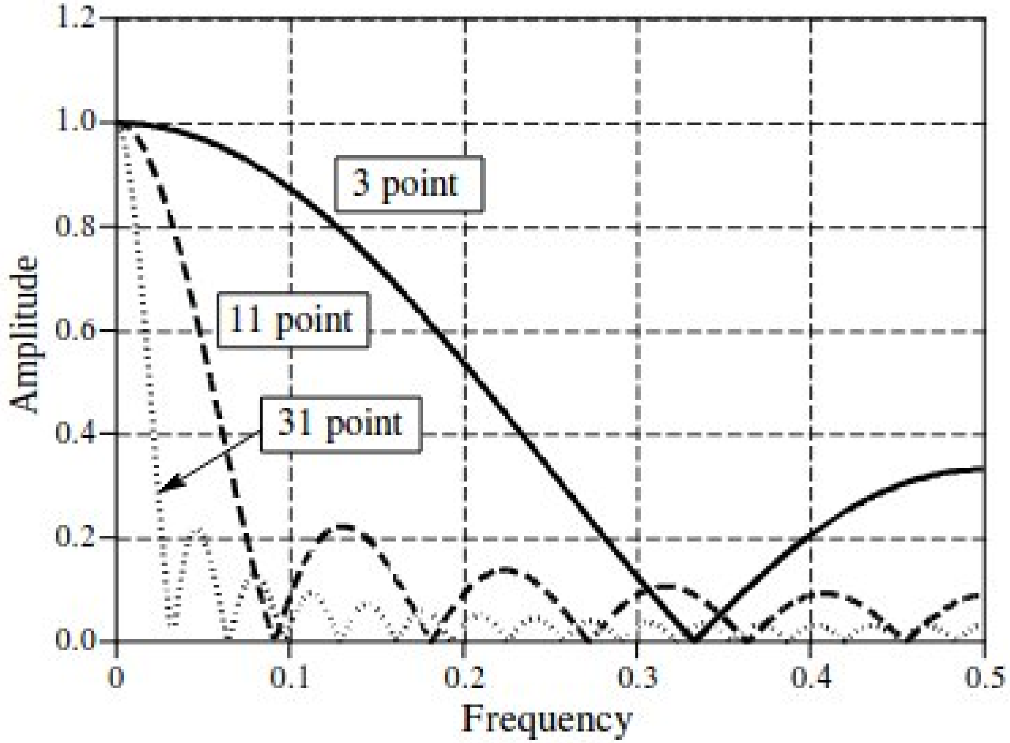
- ren midling
- ingen math omkring samplingsfrekvens mm
- kun
- samples der midles

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j]$$

```c
int dataAr[NRPINS];

int dataIndex=0;
int dataSum = 0;

void initSMA()
{
        for (int i = 0; i < NRPINS; i++)
        {
                dataAr[i] = 0;
        }
}

int calcSMA(int newSample)
{

        dataSum -= dataAr[dataIndex];
        dataSum += newSample;

        dataAr[dataIndex] = newSample;

        dataIndex++;
        dataIndex %=NRPINS;

        return  dataSum/NRPINS;
}

int main()
{
        int filterV;
        while (1) {
        // wait on right time to sample

        filterV = calcSMA(readAnalogPort(123));

        // do something :-)
        }
}
```

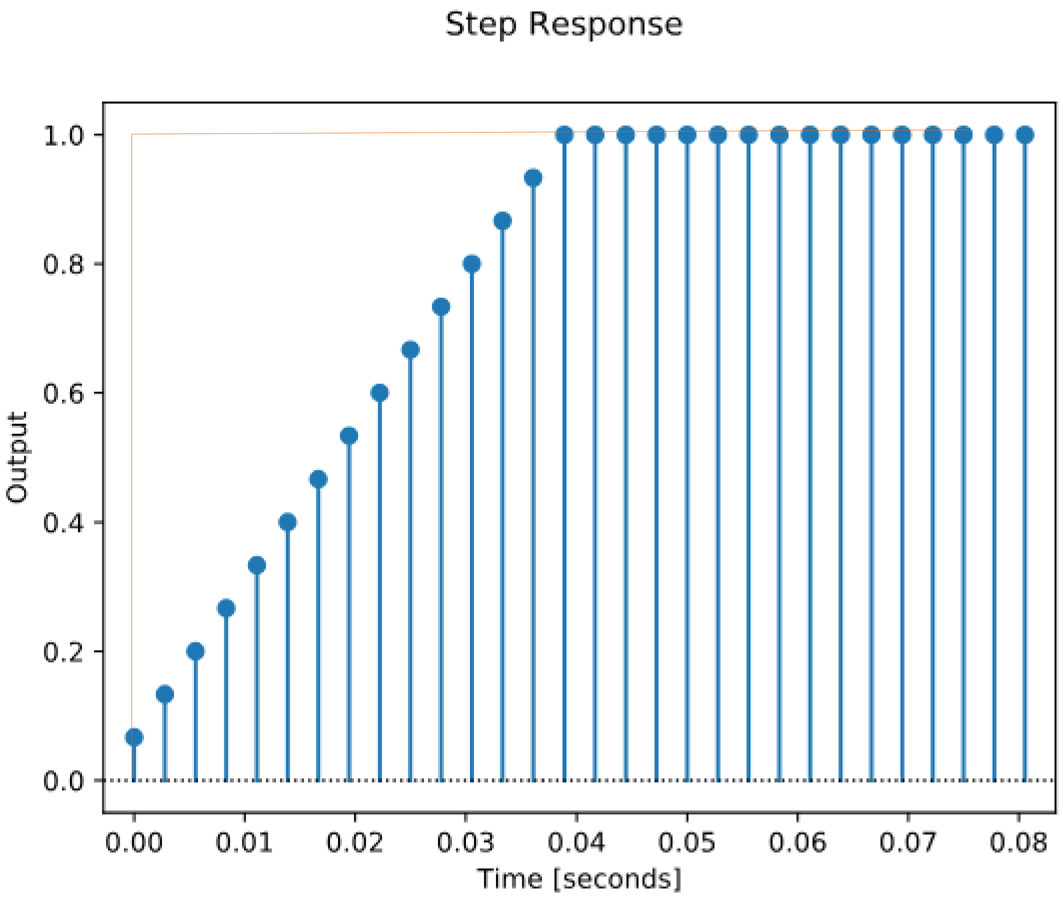
lavpas virkning af SMA

FIGURE 15-2
Frequency response of the moving average filter. The moving average is a very poor low-pass filter, due to its slow roll-off and poor stopband attenuation. These curves are generated by Eq. 15-2.

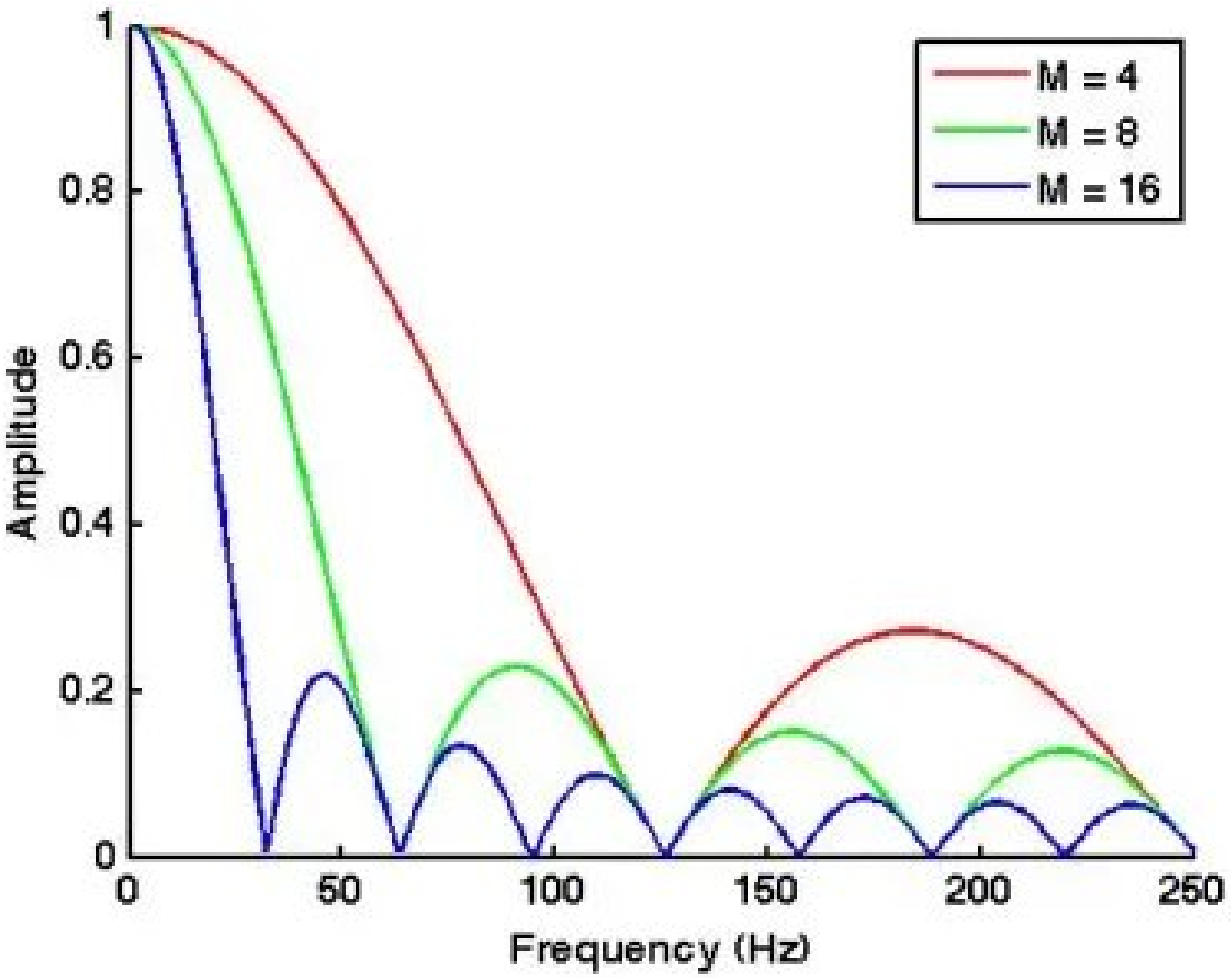


step respons & 15 pins SMA

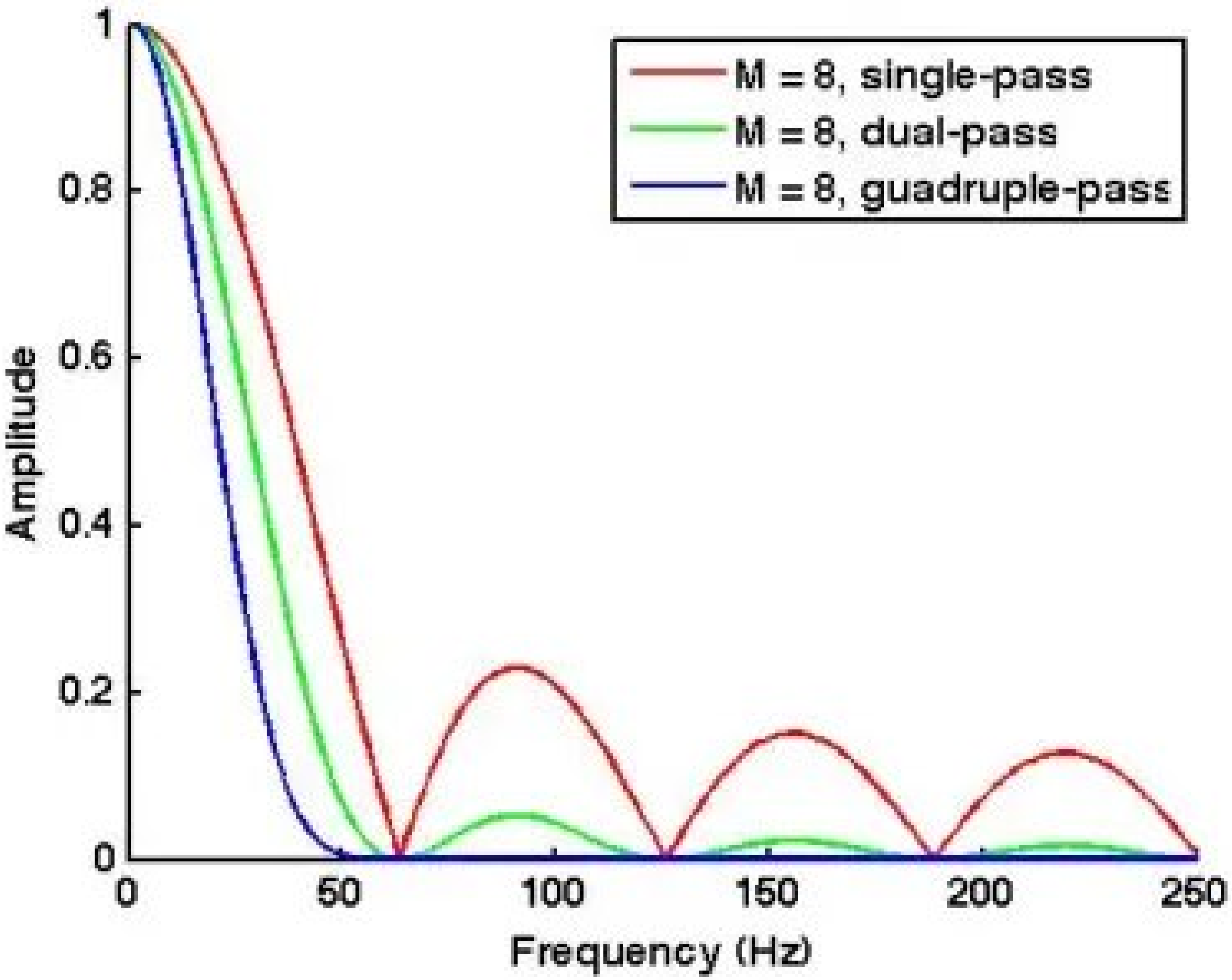
- som forventet



forskellige antal pins i SMA (Fs = 500Hz)



8 lang SMA 1,2,4 gange på samme data (Fs=500Hz)



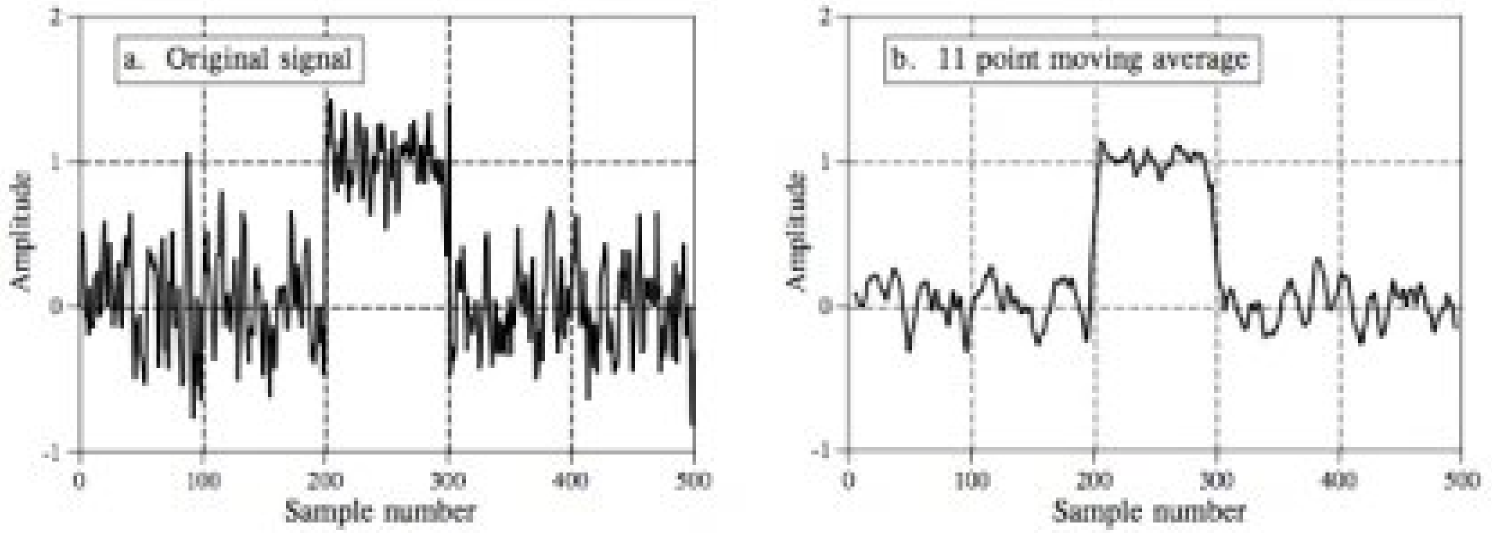
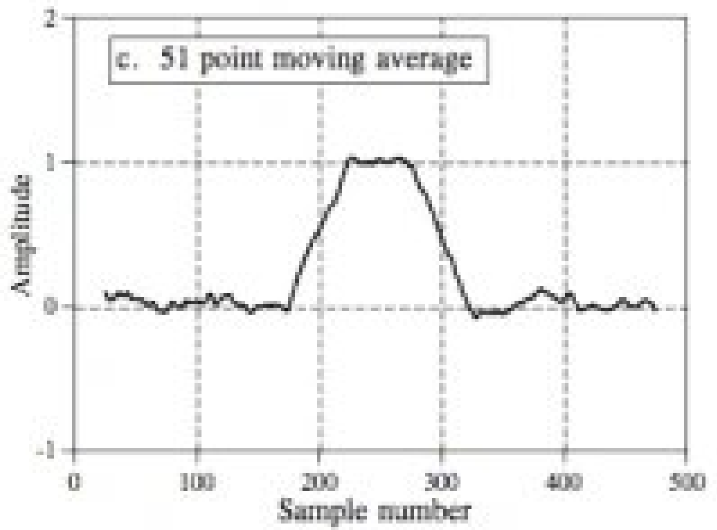


FIGURE 15-1
Example of a moving average filter. In (a), a rectangular pulse is buried in random noise. In (b) and (c), this signal is filtered with 11 and 51 point moving average filters, respectively. As the number of points in the filter increases, the noise becomes lower; however, the edges becoming less sharp. The moving average filter is the *optimal* solution for this problem, providing the lowest noise possible for a given edge sharpness.



Expotential Moving Average (EMA)

- egentlig std digital filter

- Tuning vha alpha
- alpha = 1 : ingen filter
- alpha -> 0 strong filter

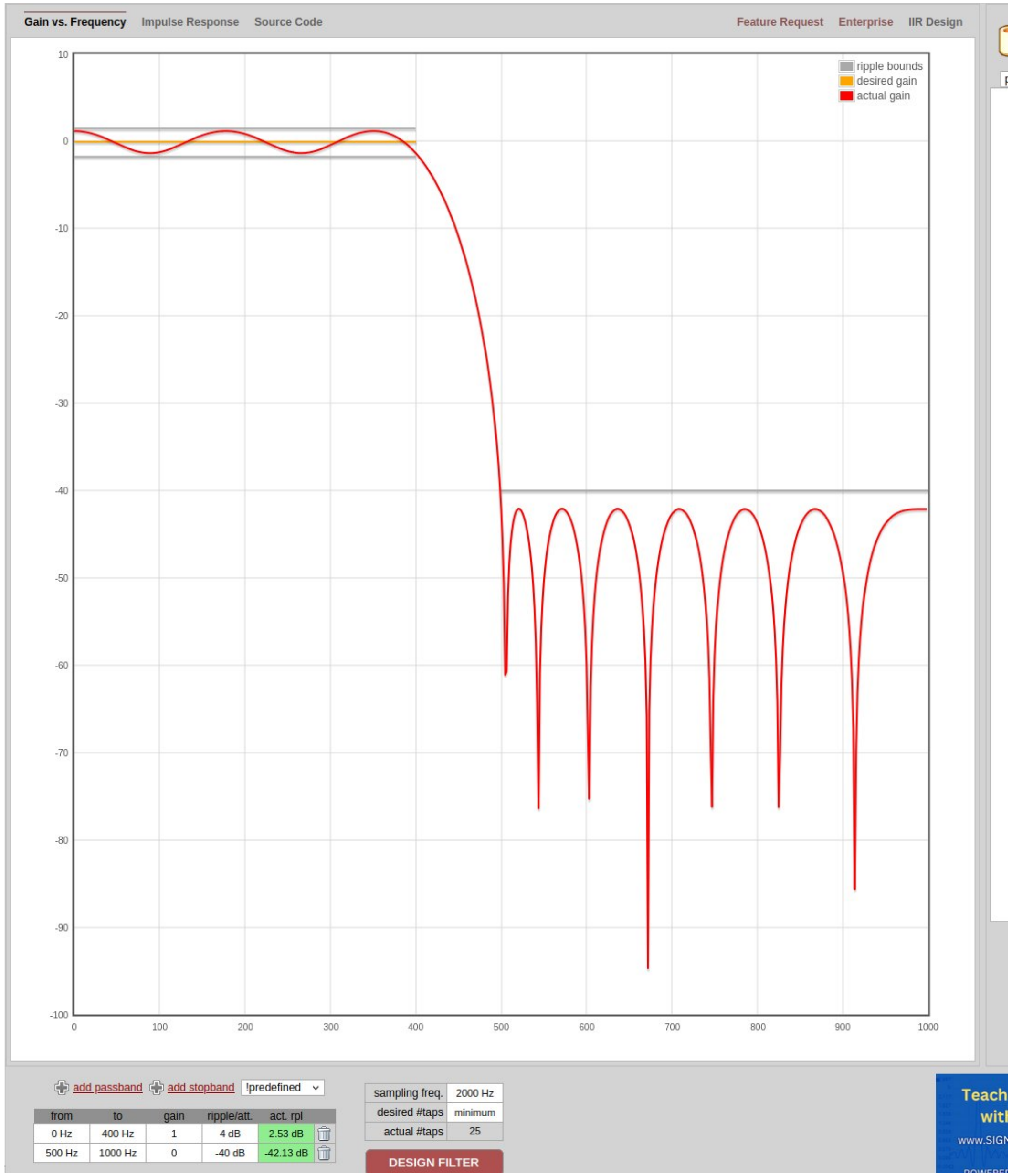
$$y[n] = \alpha x[n] + (1 - \alpha) y[n-1]$$

- Test it on digitized signal (in matlab,python,...)
- Do FFT analysis or at least visual inspection

- Der

Tools

- High order filters :== stejlere afskæring af højfrekvens



http://t-filter.engineerjs.com/

- lets see it in work

det var dagens appetizer