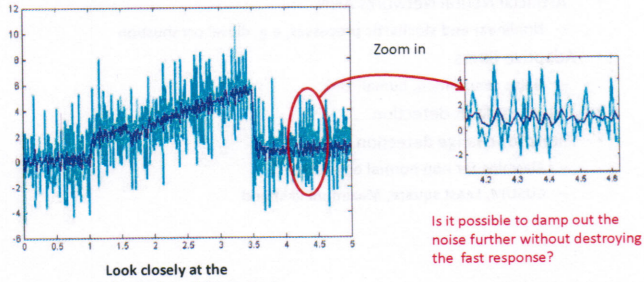
Example: try a faster lowpass filter

- Again look at the time scale at 3.5 s, try a Lp filter with $\tau \leq 0.1$ s, ex, $\omega_c = 50$ rad/s
- Now we see something looking like a step at 3.5 s and some further response between 1.0 and 2.0 s
- The filtered signal is now however noisy as well.

$$G_{LP2}(s) = \frac{50}{s+50}$$

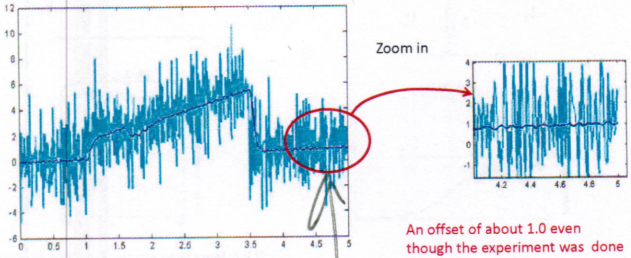


Look closely at the signal where it is constant

Is it possible to damp out the noise further without destroying the fast response?

Example: try same filter with higher order

- You should try different orders 2,3...
- Here we just test with a fourth order version $G_{LP3}(s) = \left(\frac{50}{s+50}\right)^4$



Now it looks pretty good! But there seems to be a non zero value at 5 s.

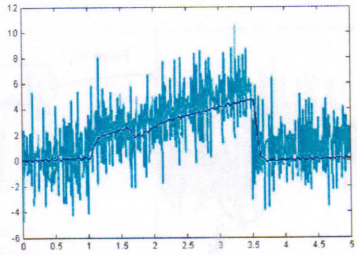
An offset of about 1.0 even though the experiment was done such that the signal should be zero. What should we do?

Drift because of integrator.

≈0.8 drift over 5 s => $\tau \approx 0.1$

Example: extend the LP filter with a highpass

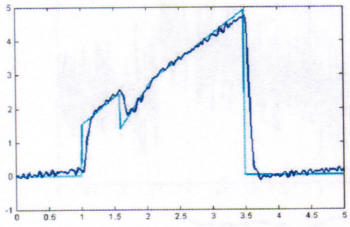
- Start with a very low cutoff frequency not to destroy some of the real low frequency contents in the signal
- Try: $G_{HP}(s) = \frac{s}{s+0.1}$



Now it is almost zero at 5 s, perhaps a bit faster.

Example: how good were we?

- In this made up example we have the true signal to compare with.
- In a real world example we do not have the answer like this.
- Other criteria then become important.
  - How should the signal be used? In a feedback loop -> what is the required dynamic.
  - As a gauge for a human to look at, what is important? Fast response or noise free.

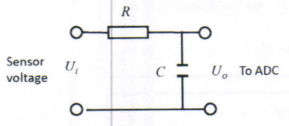


Anti aliasing filters

- Real signals are not limited in frequency. High frequency components must be filtered away with an analog lowpass filter to avoid aliasing.
- Normally a first order filter is sufficient
- How do you select the cutoff frequency of the filter?
  - Theoretically a filter with $\omega_c = \omega_s/2$ and infinitely high roll of gain would do the job.
  - As rule of thumb a first order filter with $\omega_c = \omega_s/3...6$ is sufficient.
- Design of analog filters based on a specification from a T.F *G(s)*.
  - Passive filter design
  - Active filter design

First order lowpass filter

Can be realized as a capacitor resistor or inductor resistor circuit

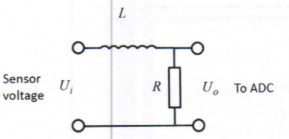


Consecutive equations:

$$C\frac{dU_C}{dt} = i$$
$$U_R = Ri$$

Loop equation

$$U_R + U_C = U_i$$
$$RCsU_C + U_C = U_i$$
$$U_C = U_o = \frac{1}{RCs+1}U_i$$



Consecutive equations:

$$L\frac{di}{dt} = U_L$$
$$U_R = Ri$$

Loop equation

$$U_R + U_L = U_i$$
$$U_o + Lsi = U_i$$
$$U_o + \frac{Ls}{R}U_o = U_i$$
$$U_o = \frac{1}{L/Rs+1}U_i$$

Dynamics of Lowpass filter

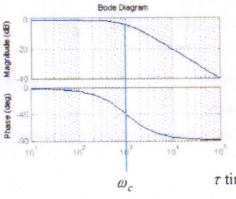
Filter	Time constant (s)	Cutoff frequency (rad/s)	Dc-gain
$U_o = \frac{1}{RCs+1}U_i$	$\tau = RC$	$\omega_C = \frac{1}{RC}$	1.0
$U_o = \frac{1}{L/Rs+1}U_i$	$\tau = \frac{L}{R}$	$\omega_C = \frac{R}{L}$	1.0

Exampel with

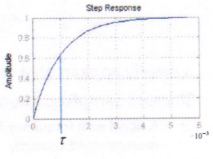
$R = 10\,\text{k}\Omega$

$C = 100\,\text{pF}$

$\omega_C = \frac{1}{RC} = 1000\ \text{rad/s}$



ω_c



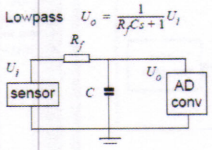
τ

τ time where the amplitude is 63% of max gain

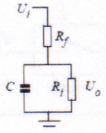
Input impedance of the ADC

The components must be selected based on the input resistance of the ADC

Lowpass $U_o = \frac{1}{R_f Cs+1}U_i$



R_i is the input resistance of the AD-conv. Typically around 1MΩ



$$U_o = \frac{R_i}{R_i R_f Cs + R_i + R_f}U_i$$

dc-gain $\frac{R_i}{R_i + R_f}$

time constant $\tau = \frac{R_i R_f C}{R_i + R_f}$

- Example with R_i = 1MΩ, τ = 1ms
- Select Capacitor -> C = 100pF
- gives R_f = 10kΩ and dc-gain = 0.99

Want $R_i >> R_f$

Lecture Outline

- Introduction to filters
- Standard filters
  - Continuous time, frequency domain design, Laplace
- Approximating continuous time filters to discrete time
  - Z-transform vs. Laplace transform
- **Direct discrete time filter design**
  - IIR and FIR filters
- Digital implementation of filters
  - Sampling of signals
- Analog implementation of filters
  - passive filters
  - active filters
- Example: noisy sensor

Direct digital filter design using Matlab (IIR)

- The Signal processing toolbox in Matlab gives many possibilities to design digital filters directly based on specifications. Both command line and graphical.
- The standard filter is called Butterworth, others are
  - Chebychev, Eliptict, Bessel etc.
  - They are Infinite impulse response, IIR filters.
- Passband and stop band frequencies, ω_c are given as normalized frequencies, $0.0 < \omega_c < 1.0$, where 1.0 corresponds to half the sampling frequency.
- Example: same highpass filter as before

Extract of Matlab code for,

$$G(s) = \frac{s}{s+100}$$

with $T_s = 0.005$

```
>> wc = 100;
>> Ts = 0.005;
>> ws = 2*pi/Ts;
>> wn = wc/(ws/2)
wn =
   0.1592
>> [num,den]=butter(1,wn,'high')
num =
   0.7966  -0.7966
den =
   1.0000  -0.5932
>> Gz=tf(num,den,Ts)

Transfer function:
0.7966 z - 0.7966
-----------------
  z - 0.5932

Sampling time: 0.005
```

Lecture Outline

- Introduction to filters
- Standard filters
  - Continuous time, frequency domain design, Laplace
- Approximating continuous time filters to discrete time
  - Z-transform vs. Laplace transform
- Direct discrete time filter design
  - IIR and FIR filters
- **Digital implementation of filters**
  - Sampling of signals
- Analog implementation of filters
  - passive filters
  - active filters
- Example: noisy sensor

Code structure

1. Generate a timer interrupt to execute the code patch (function) for the filter calculations each T_s second.

Code

2. Read the sampled sensor value from the A/D converter
3. Scale the sensor value to the right unit.
4. Calculate the filter output -> difference equation
5. Shift the filter states.
6. Do something else
7. Wait until next timer interrupt

Loop

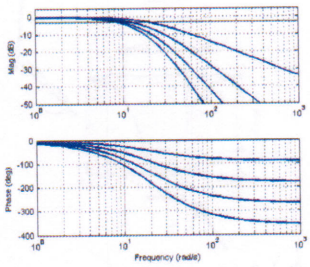
Higher order lowpass filter

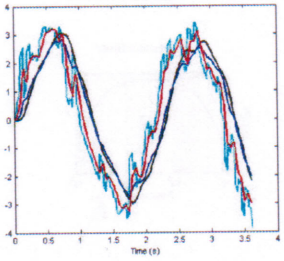
For the same $\omega_c = 20$ as in last slide we increase the order by taking

$$G_1 = \frac{\omega_c}{s+\omega_c}, G_n = G_1^{\,n}, \text{ with } n = 1\ldots 4$$

The roll off frequency increases with 10 dB per decade and the phase with - 90 deg for each order of n.

Example:
Cyan line noisy sensor
Blue line: ω_c = 5 rad/s, *n=1*
Red line: ω_c = 20 rad/s, *n=1*
Black line: ω_c = 20 rad/s, *n=4*





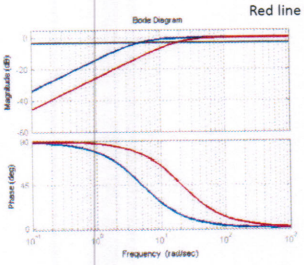
Highpass filter

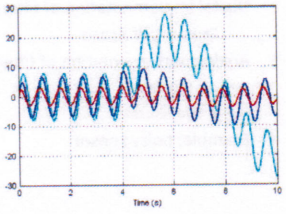
$$G_{HP}(s) = \frac{s}{s+\omega_c}$$

What are the gains at Zero and infinite frequency?

Remove slowly moving offsets,
Example: Integrating a rate gyro to get the angle if the rate gyro noise is not zero mean will the integrated angle drift away.

Example:
Cyan line sensor with drift at time 4 s
Blue line filtered sensor with ω_c = 5 rad/s
Red line filtered sensor with ω_c = 20 rad/s

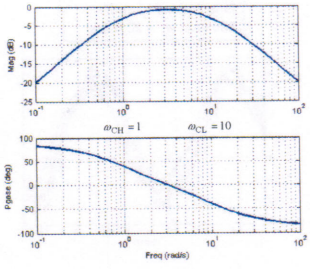


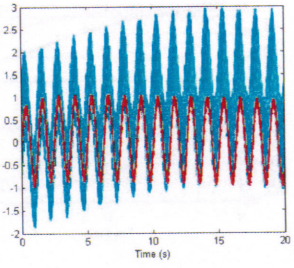


Bandpass filter

Simply combine one low pass and one high pass filter in series, where the cut off frequencies must be, $\omega_{CL} > \omega_{CH}$

Example:
Cyan line: $y = 1.0\sin(0.1t) + 1.0\sin(5t) + 1.0\sin(100t)$
Red line: $y_f = \frac{10s}{(s+1)(s+10)} y,\ y_f \approx 1.0\sin(5t)$



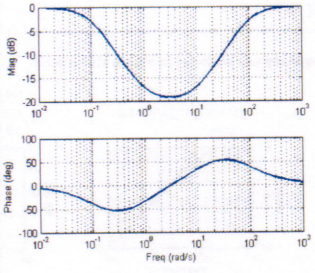


Stop band filter

A stop band filter needs at least four cutoff frequencies
If the low and high frequency gain should be 1.0

Example:

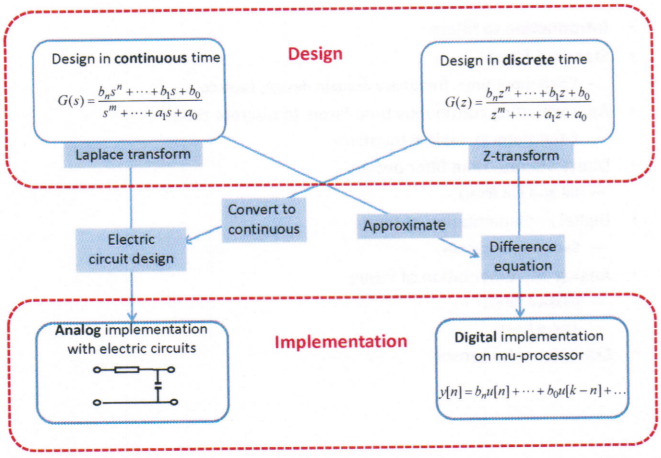
$$G_{SB}(s) = \frac{(s+1)(s+10)}{(s+0.1)(s+100)}$$



What are the high and low (dc) gains?

Can we say something about the phase?

1.)Design 2.)Implement



Continuous time transfer function: Laplace

The Laplace transform of a time serie *u(t)* is defined as:

$$L\{u(t)\}=\int_0^\infty u(t)e^{-st}dt$$

A transfer function *G(s)*, is the ratio of the output Laplace transform with the input Laplace transform.

$$G(s)=\frac{L\{y(t)\}}{L\{u(t)\}}$$
$$Y(s)=G(s)U(s)$$

Two important special cases: derivative and integration. If the intitial conditions are zero, *u(0)=0*, then:

$$L\left\{\frac{d^n u}{dt^n}\right\}=s^n$$

$$L\left\{\int_0^\infty u(t)dt\right\}=\frac{1}{s}$$

Discrete time transfer function: Z-transform

The z-transform of a time serie *u(k)* is defined as:

$$L\{u(k)\}=u(z)=\sum_{k=0}^{\infty}u(k)z^{-k}$$

A transfer function *G(z)*, is the ratio of the output z-transform with the input z-transform.

$$G(z)=\frac{L\{y(k)\}}{L\{u(k)\}}$$
$$Y(z)=G(z)U(z)$$

The z-transform is related to the Laplace transform as.

$$z=e^{sT_s}$$

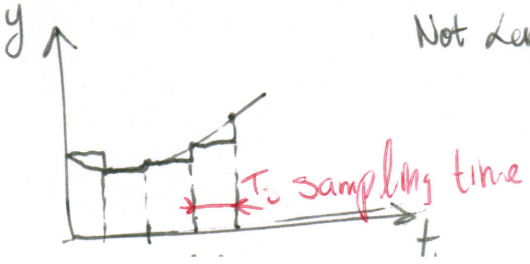
An important property of the z-transform is the **shift property**

$$L\{y(k+n)\}=z^n y(k)$$

Example:

$$L^{-1}\{z^n y(z)\}=y[k+n]$$
$$L^{-1}\{z^{-n} y(z)\}=y[k-n]$$

Not Laplace L



Pros and cons: Discrete vs. Continuous design

- Design of continuous time filters are typically done based on understanding the filters effect on a signal in time domain.
  - Very quick and easy method for lower order filters
  - The design can easily be done directly with electric components for an analog implementation.
- Discrete time filters are typically designed using computer tools (Matlab)
  - No difference in effort designing low or high order filters, e.g., order of twenty or higher.
  - The filter type must be decided in advance (Butter, Chebychev, etc.)
  - The effect of sampling frequency of the filtered signal must be known
  - FIR filters can be designed, they are filters with only numerator, no denominator