Robust Lec 5

Designing and implementing filters in mechatronic systems

Bengt Eriksson
benke@md.kth.se

Active filter: External power supply

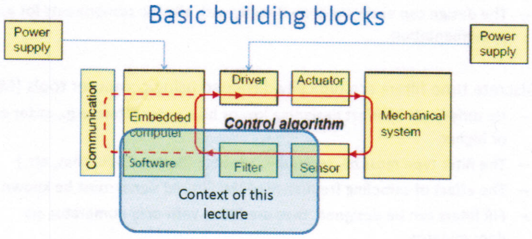
Z-transform: Discrete

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

Filters in mechatronic system

Basic building blocks



Why do we need filters in mechatronic systems?

- Change the characteristics of a signal
  - Frequency contents
  - Time sequence
- Change the dynamic characteristics of a signal
  - Integration and derivation
  - Filter out specific frequencies, bandpass and stop band filters
- Reduce the noise of a signal
  - Electric noise from disturbances
  - Quantization noise
  - Mechanical noise

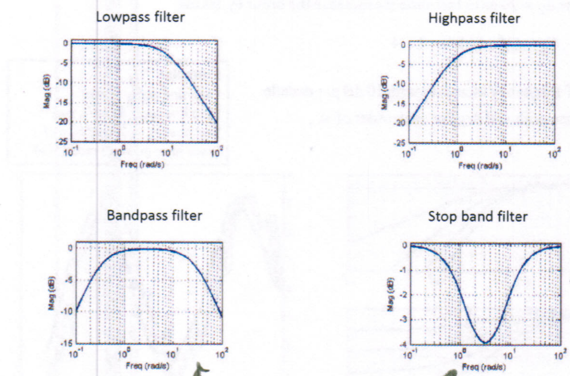
Due to resolution
Vibrations

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

The four basic filters in frequency domain

For mechanical resonance freq.



Combinations

Low frequency: A drift. Might come from sensor integration. E.g. gyros, position sensors, measure with integration.

Frequency representation: Gain and Phase

The gain, a and phase φ for any transfer function $G(s)$ with the input $u = 1.0\sin(\omega t)$ is $y = a\sin(\omega t + \varphi)$ is calculated over the complex frequency $s = j\omega$, where:

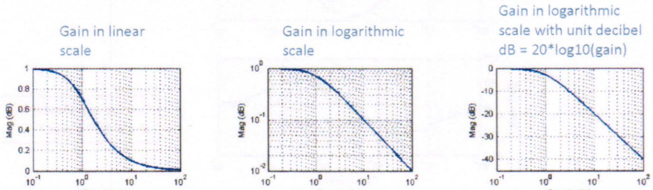
$a = \sqrt{\mathrm{real}G(j\omega)^2 + \mathrm{imag}G(j\omega)^2}$ and

the phase is the angle between the real and complex parts as:

$$\tan\varphi = \frac{\mathrm{imag}G(j\omega)}{\mathrm{real}G(j\omega)}$$

How does the Phase look like for this filter?

The frequency axis is always in a logarithmic scale



-90° phase

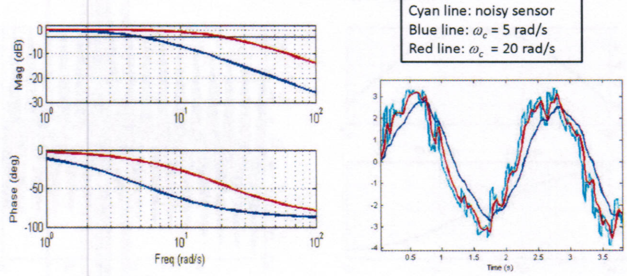
1:st order lowpass filter

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

What are the gains at Zero and infinite frequency?

ω_c is the cutoff frequency, which is defined as:
The frequency when the gain is -3 dB
Lower ω_c gives higher noise reduction
Lower ω_c gives larger delay of the filter output

Example:
Cyan line: noisy sensor
Blue line: ω_c = 5 rad/s
Red line: ω_c = 20 rad/s

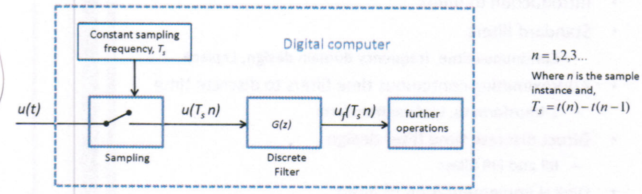


Choise of phase lag and feedthrough noise

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

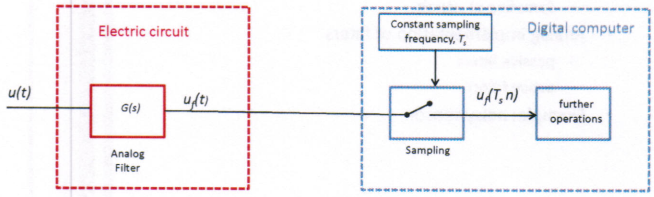
Digital versus analog implementation



$n = 1,2,3...$

Where n is the sample instance and,

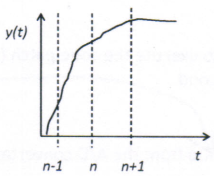
$T_s = t(n) - t(n-1)$



Approximations of continuous time transfer functions

Sampling period

$T_s = t(n) - t(n-1)$



Shift property with the z - operator

forward shift $z \cdot y \Rightarrow y(n+1)$

backward shift $z^{-1} \cdot y \Rightarrow y(n-1)$

$$\frac{dy}{dt} = sy \approx \frac{y[n+1]-y[n]}{T_s} = \frac{z-1}{T_s} y$$ Euler forward

$$\frac{dy}{dt} = sy \approx \frac{y[n]-y[n-1]}{T_s} = \frac{1-z^{-1}}{T_s} y = \frac{z-1}{zT_s} y$$ Euler backward, Backward difference

$$\frac{dy}{dt} = sy \approx \frac{2}{T_s}\frac{y[n+1]-y[n]}{y[n+1]+y[n]} = \frac{2}{T_s}\frac{z-1}{z+1} y$$ Tustin, Trapezoidal, Bilinear

Numeric approximation using matlab

```
>> s = tf('s')                          Define the Laplace operator

Transfer function:
s

>> G = 50/(s+50)                        Give the continuous time transferfunction

Transfer function:
 50
------
s + 50

>> Ts = 0.005;                          Give a sample time, 10 to 30 times faster than the filter
>> Gd = c2d(G,Ts,'tustin')              Calculate the discrete time transferfunction

Transfer function:
0.1111 z + 0.1111
-----------------
  z - 0.7778

Sampling time: 0.005
>>
```

For the lab

Example: implementing the highpass filter

1. Rewriting the highpass filter $y_f(z) = \frac{0.8z - 0.8}{z - 0.6} y(z)$ by multipliing with z^{-1}

$$y_f = \frac{0.8 - 0.8z^{-1}}{1 - 0.6z^{-1}} y \quad \text{then} \quad (1 - 0.6z^{-1}) y_f = (0.8 - 0.8z^{-1}) y$$

2. Convert to difference equation using the shift property

$$y_f[n] = 0.6 y_f[n-1] + 0.8 y[n] - 0.8 y[n-1]$$

3. Write the code

Code

1. Sample:
2. Calculate:
3. Shift:

```
y = read_adc1();
yf = 0.6*yf_old + 0.8*y - 0.8*y_old;
yf_old = yf; y_old = y;
```

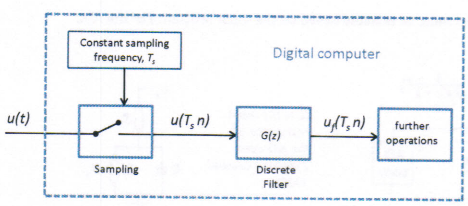
Observe that `yf_old` and `y_old` must be declared static (non-volatile)

Not good to declare vars as globals.

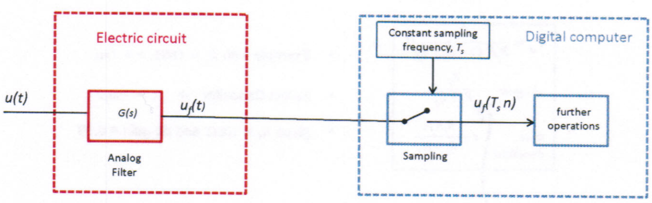
Lecture Outline

- 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
- Exaxmple: Noisy sensor

Digital versus analog implementation

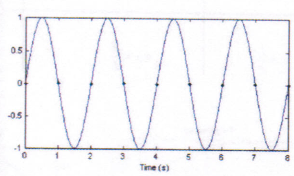


$n = 1,2,3...$
Where n is the samle instance and,
$T_s = t(n) - t(n-1)$



Why use analog filters

- The main reason to use an analog filter is because of Aliasing effects from sampling.
- What is Aliasing?
- An example: A sine wave signal with frequency 0.5 Hz is sampled with the frequency 1.0 Hz

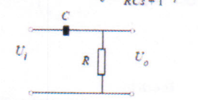


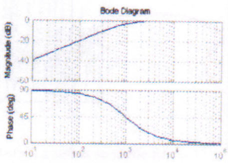
One sample per second at the black dots.

- It clearly appears that sampling at this frequency gives nothing (0.0 Hz)
- Shanons samplings theorem:
  - A signal with frequency f_s must be sampled with a frequency at least $2f_s$ to be reconstructed correctly

Other filters

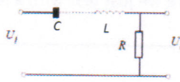
Highpass $U_o = \frac{RCs}{RCs+1}U_i$



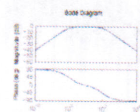


Same parameters as in the lowpass filter $\omega_C = 1000$ rad/s

Bandpass filter



$$U_o = \frac{(R/L)s}{s^2+(R/L)s+1/(LC)}U_i$$



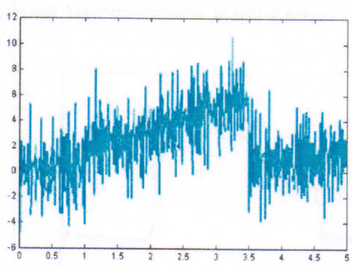
If R is large then are the cutoff frequencies for the derivative and integral parts almost the same as for the individual filters

Filters based on statistical properties and other methods

- Kalman filter
  - can include a model of the filtered property
  - Satellite tracking, navigation, computer vision
- Artificial Neural Networks ANN
  - Nonlinear and stochastic processes, e.g. diesel combustion
- Adaptive filters
  - Noise cancelation, human EKG,
- Filters for fault detection
- Filters for change detection
  - Alarming for non normal operation
  - CUSUM, Least square, Maximum liklihood

Example: noisy sensor signal

- The task is to design a filter for a noisy sensor
- Step 1. record the raw signal and save it to a file so that it can be plotted and inspected in Matlab.
- Where should we start?



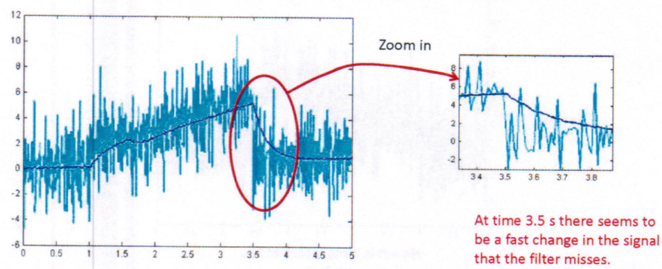
Rise time ≈ 3s

ω_c = inverse of time const.

Example: try lowpass

- Start with a low cutoff frequency to see if there is something in the signal.
- Look at the time scale, try a Lp filter with $\tau \approx 1$ s, ex, $\omega_c = 5$ rad/s

$$G_{LP1}(s) = \frac{5}{s+5}$$



Looks good but....

At time 3.5 s there seems to be a fast change in the signal that the filter misses. Try a faster filter.