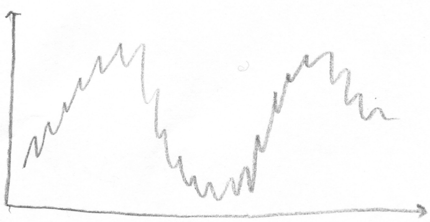
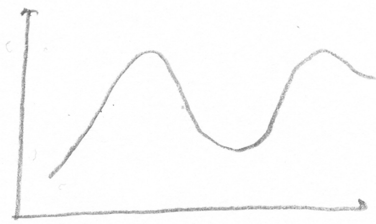
Filters

want to remove noise & unwanted signals from our signal we need filters

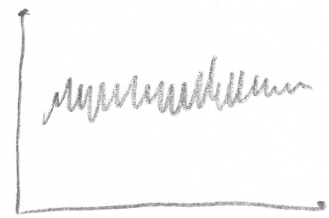
ex



measured

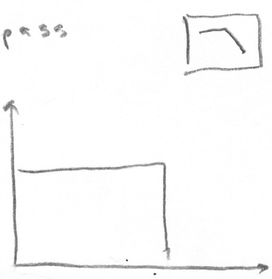


signal



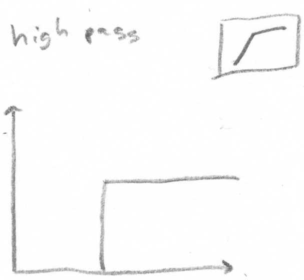
noise

low pass



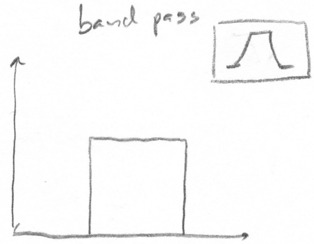
ω_0

high pass



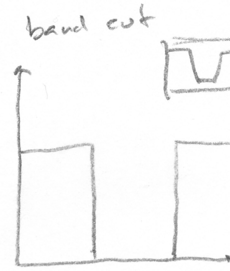
ω_0

band pass



ω_1 ω_2

band cut

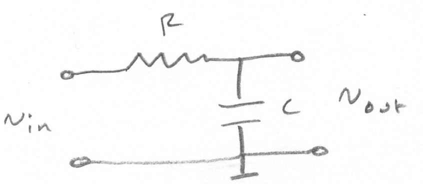


ω_1 ω_2

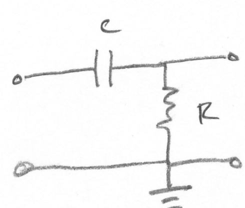
How do we create these filters with physical components

$$Z_R = R$$

$$Z_C = \frac{1}{i\omega C}$$



low pass



high pass

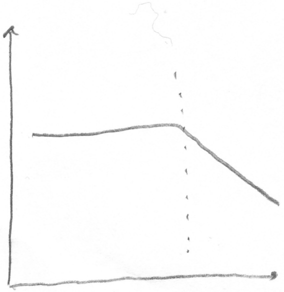
$$\frac{v_{out}}{v_{in}} = \frac{\frac{1}{i\omega C}}{R + \frac{1}{i\omega C}}$$

Laplace Transform $i\omega \to s$

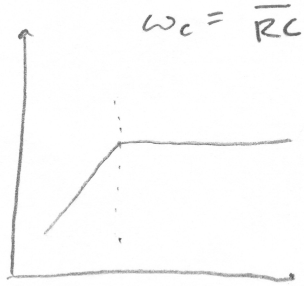
$$\omega_c = \frac{1}{RC} \; ; \; RC = \tau$$

$$H_{LP}(s) = \frac{\frac{1}{sC}}{R + \frac{1}{sC}} = \frac{1}{RCs+1}$$

$$H_{HP}(s) = \frac{RCs}{RCs+1}$$



ω_c



ω_c

Implementing a Digital Filter

moving average, 2 samples

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

what is time constant τ? depends on sample rate

if $T_s = .01$ (100 Hz) then the above has

a time constant of .02

we can generalize this to

$$y[n] = \frac{1}{N}\sum_{k=0}^{N} x[n-k]$$ FIR filter

This moving average filter weights all samples equally

we can weight older data by using the previous output

$$y[n] = \alpha\, x[n] - (1-\alpha)\, y[n-1]$$ IIR Filter

this is the difference equation for an exponential

moving average filter

it turns out if $\alpha = \frac{\Delta t}{RC + \Delta t}$ we now have a

discrete time version of our RC lowpass filter

To create a discrete time filter from a continuous

filter we need the z-transform

$$z \rightarrow e^{Ts}$$ where T is the sampling period

if we know the transfer function of a filter or system

in the s-domain we can find a close approximation

for our discrete system in the z-domain

Example: highpass filter at 10 Hz sample rate is 100 Hz

$$\tau = \frac{1}{2\pi f} = 0.0159$$

$$H(s) = \frac{\tau s}{\tau s + 1} = \frac{0.0159\, s}{0.0159\, s + 1}$$

$$G(z) = \frac{0.7609\, z - 0.7609}{z - 0.5219}$$

DSP format is

$$F(z) = \frac{0.7609 - 0.7609\, z^{-1}}{1 - 0.5219\, z^{-1}}$$

Difference equation is then:

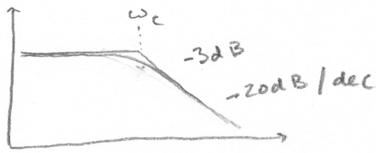
$$y[n] = 0.7609\, x[n] - 0.7609\, x[n-1] + 0.5219\, y[n-1]$$

Robotis cape lib includes:

- 1st order lowpass / high pass → don't trust implementation
- moving average
- butterworth (2nd order) low & high pass
- Integrator / Double Integrator
- PID Filter

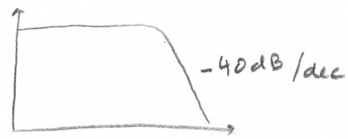
More about filters

First order filter

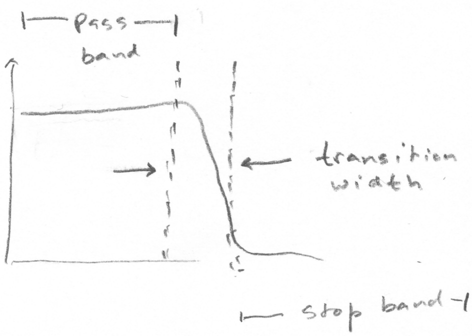


$$\frac{1}{s+1}$$

2nd order filter



$$\frac{1}{s^2+s+1}$$



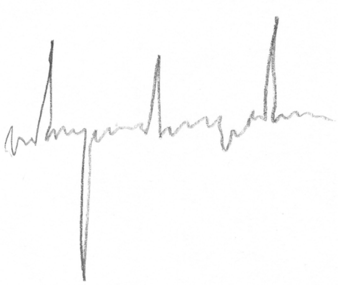
butterworth: maximally flat pass band

chebyshev: flat pass OR stop band (not both) steeper transition

elliptical: steepest transition, ripple in both pass & stop.

Other filters that are useful:

Median filter → statistical, non-linear



$$y[n] = \text{Median}\left(x[N] \ldots x[N-K]\right)$$

throws out outliers, similar to moving average.