


Filtterisuunnittelu

Signaalin muokkaus

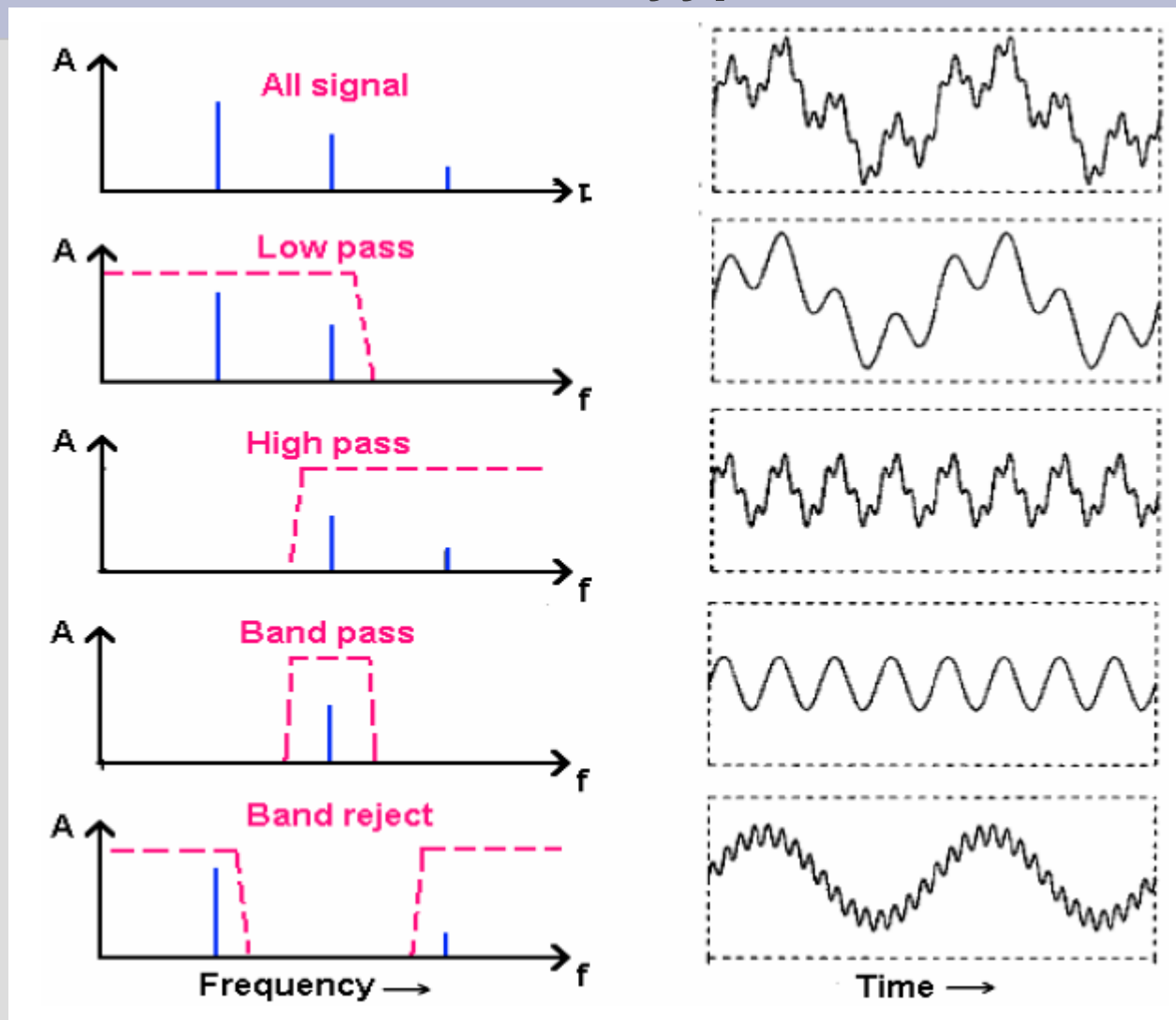
Suodatusta käytetään yleensä signaalin muokkaukseen siten, että

- poistetaan häiritsevä signaali hyötysignaalin joukosta
- vähennetään kohinan vaikutusta hyötysignaaliin
- muokataan taajuuskäyttäytymistä haluttuun suuntaan (vahvistus ja/tai vaihesiirto)
- rajoitetaan signaalin kaistanleveyttä
- ...

- Suotimien ominaisuudet halutaan yleensä esittää taajuustasossa, jolloin ollaan kiinnostettu suotimen *vahvistuksen* ja *vaihekulman* käyttäytymisestä taajuuden funktiona  BODE
- Usein halutaan suunnitella suodin, jolla muokataan vahvistusta ja/tai vaihetta eri taajuuksilla halutulla tavalla

Filterisuunnittelu

Suodintyytit



Filtterisuunnittelu

ja prosessien säätö

- Kohinan lähteet
 - mittauskohina
 - korkeataajuista suhteessa prosessiin
 - sähköhäiriöt, näytteenotto, kvantisointivirheet, ...
 - prosessikohina
 - matalataajuista
 - mikä on häiriötä, mikä kohinaa.. ?
 - kuplat, roiskeet, kanavoituminen, ...
- Filtteri
 - estää ohjauksen värinää
 - lisää viivettä (vaiheenjättöä)
 - suunnittelu on tasapainottelua

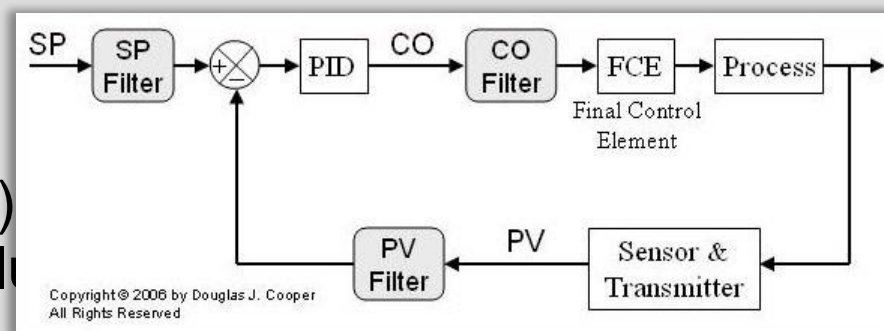
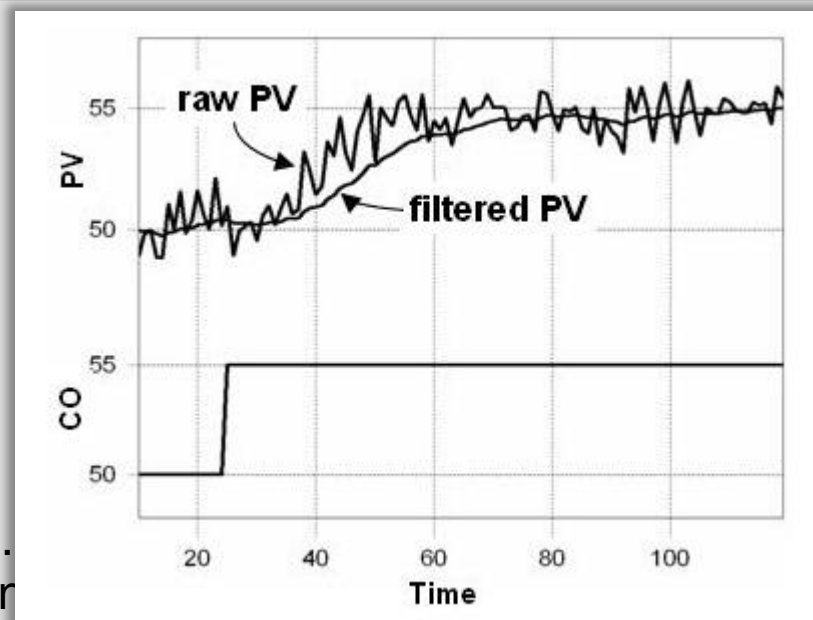
Filterisuunnittelu

ja prosessien säätö

- Kohinan lähteet
 - mittauskohina
 - korkeataajuista suhteessa prosessiin
 - sähköhäiriöt, näytteenotto, kvantisointivirheet, ...
 - prosessikohina
 - matalataajuista
 - mikä on häiriötä, mikä kohinaa..
 - kuplat, roiskeet, kanavoituminen

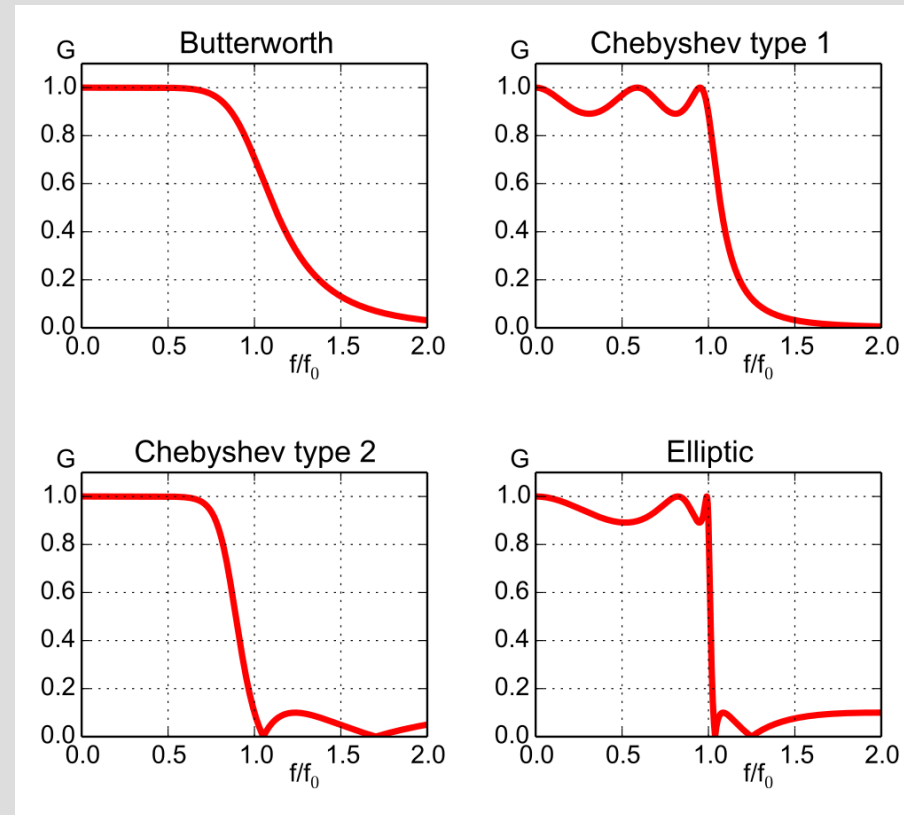
...

- Filteri
 - estää ohjauksen värinää
 - lisää viivettä (vaiheenjättöä)
 - suunnittelu on tasapainottelu



Butterworth filteri

- Butterworth properties
 - monotonic amplitude response (no ripple)
 - quick roll-off around cutoff frequency (improves with order)
 - overshoot and ringing in step response (worsens with order)



Filterisuunnittelu

Butterworth-suodin

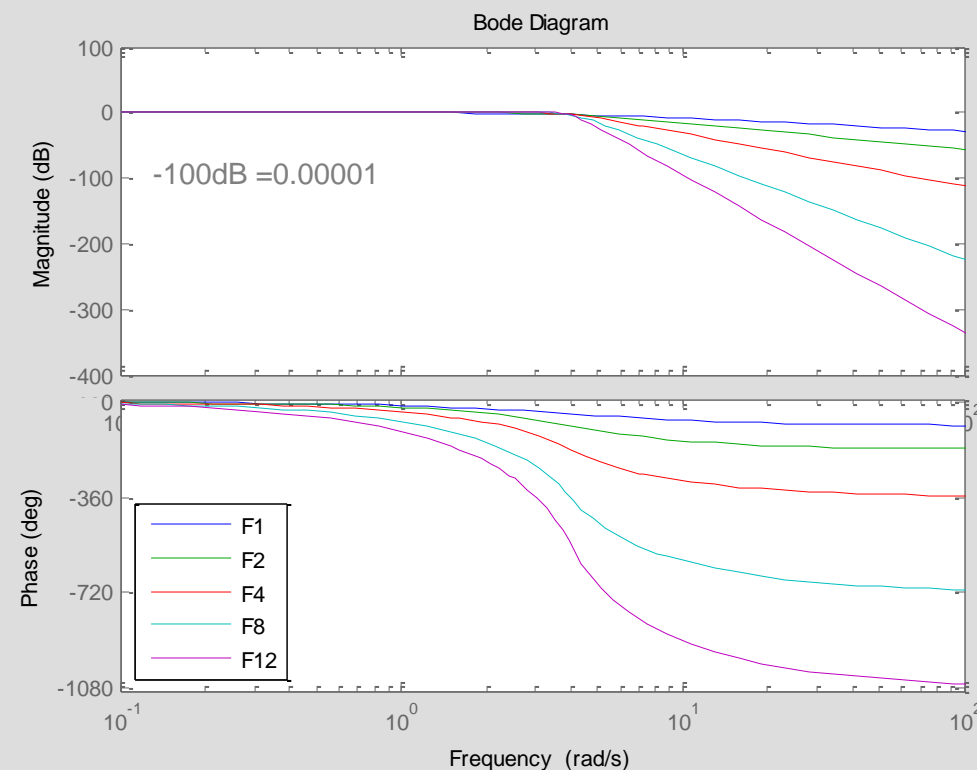
- Suunnitellaan Butterworth-filttereitä
 - alipäästösuotimia
 - kulmataajuus 4 rad/s
 - eri kertaluvulla
 $n = \{1, 2, 4, 8, 12\}$

- Matlabilla:

```

wn = 4
n = 1 % 2, 4, 8, 12
[b, a] = butter(n, wn, 'low', 's')
F1 = tf(b, a);
bode(F1)
step(F1);
  
```

- Butterworth-suotimen vahvistuskuvaaja on hyvä, mutta vaihekuvaaja heikko.

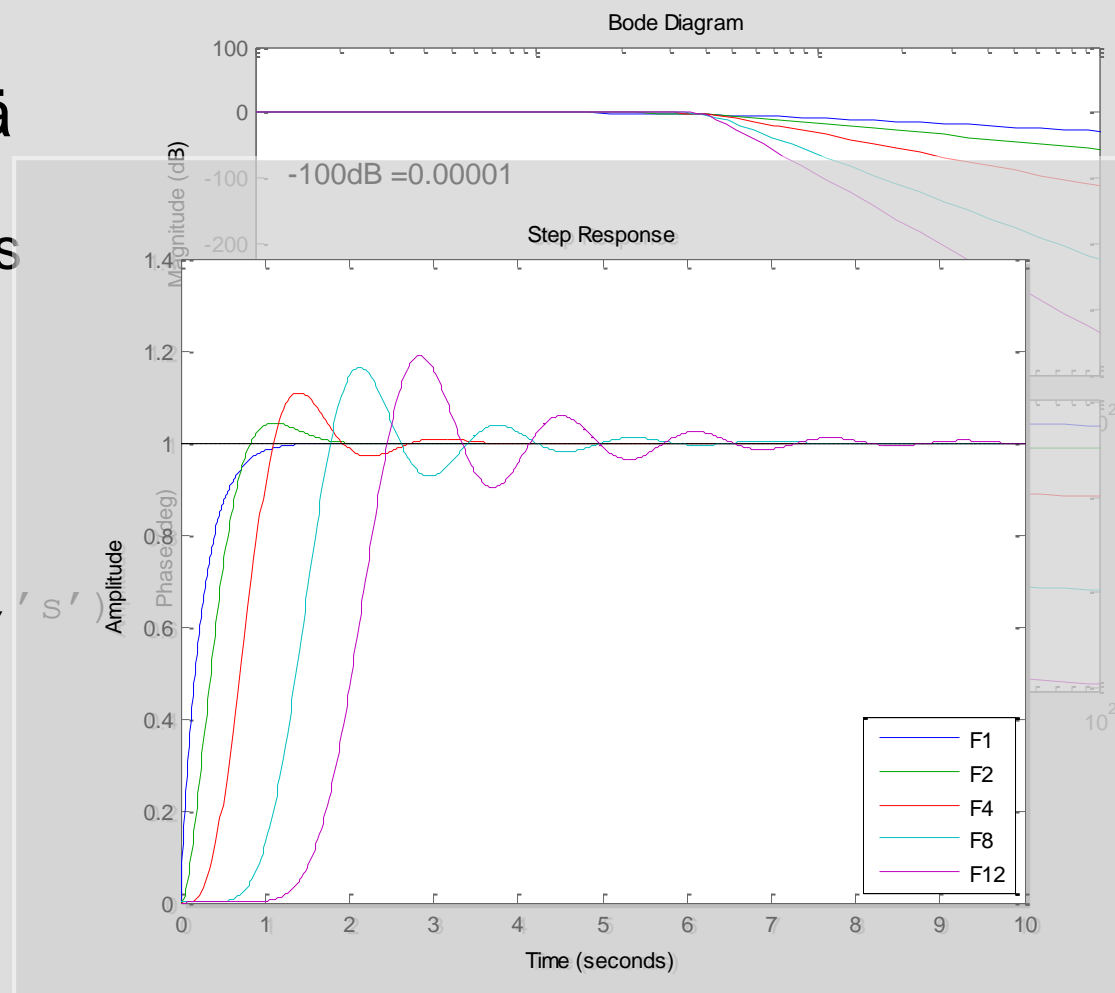


Filterisuunnittelu

Butterworth-suodin

- Suunnitellaan Butterworth-filttereitä
 - alipäästösuotimia
 - kulmataajuus 4 rad/s
 - eri kertaluvulla $n = \{1, 2, 4, 8, 12\}$
- Matlabilla:


```
wn = 4
n = 1 % 2, 4, 8, 12
[b, a] = butter(n, wn, 'low', 's');
F1 = tf(b, a);
bode(F1);
step(F1);
```
- Butterworth-suotimen vahvistuskuvaaja on hyvä, mutta vaihekuvaaja heikko.



Harjoitus: Filterisuunnittelu

a) Suunnittele seuraavat Butterworth-suotimet

- alipäästösuodin (low-pass)
- ylipäästösuodin (high-pass)
- kaistanestosuodin (stop)
- kaistanpäästösuodin (band-pass)

kun leikkaustaajuus on 0.5Hz
(kaistanesto ja -päästö +-20
leikkaustaajuuden molemmin puolin).

Voit käyttää suunnitteluun Matlabin
butter-komentoa:

```
>> [num,den]=butter(n,wn,'low','s')  
jne. kts. >> help butter
```

b) Piirrä **taajuusvaste** ja **simuloi** signaalin suodatusta aikatasossa niin että näet suotimien toimivan ok.

c) **Kokeile** suotimelle eri astelukuja, esim $n=\{3,5,12\}$.

Butterworth simulointikoodia

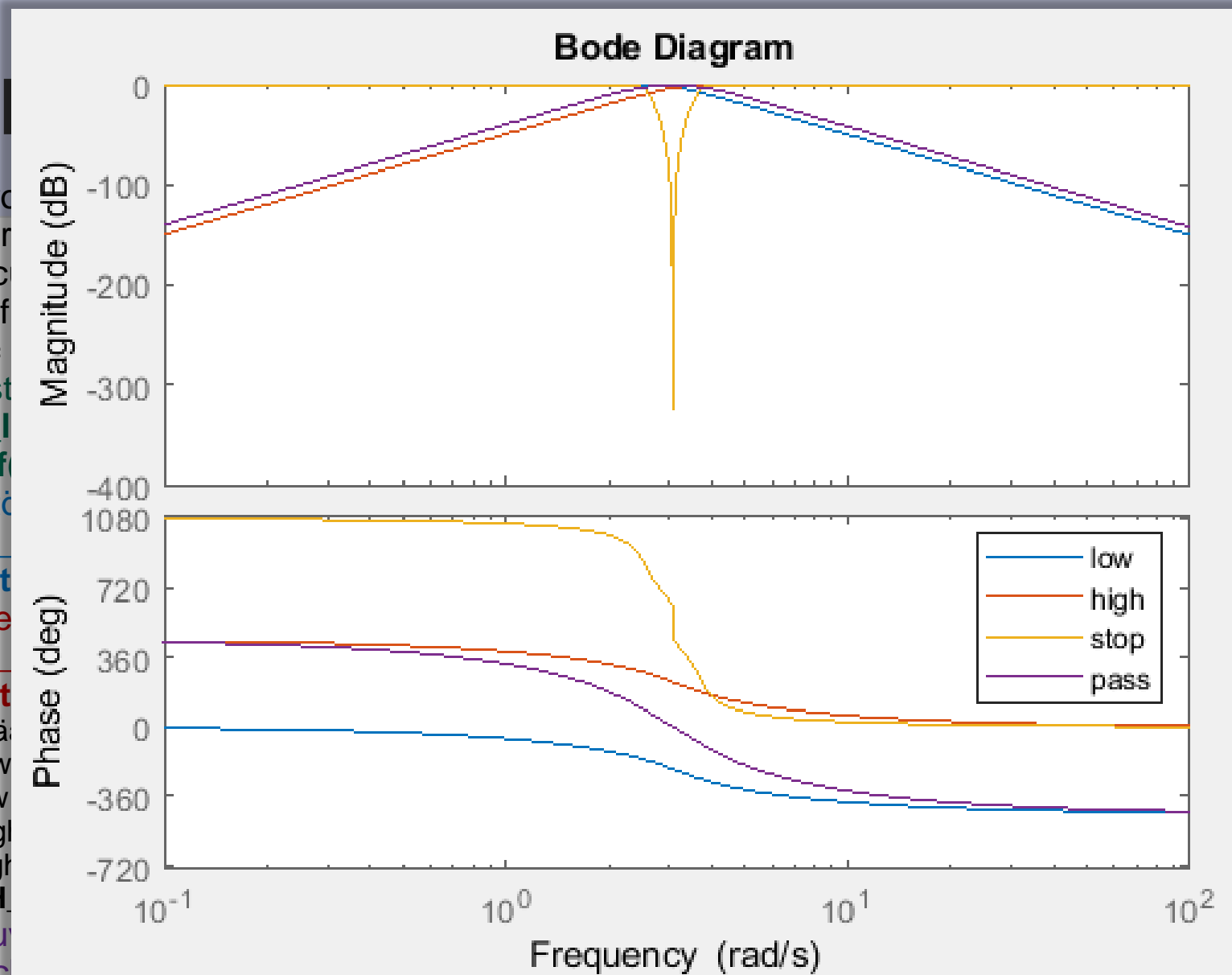
```
% Butterworth-suotimen parametrit
n = 5; % order
f = 0.5 % cutoff [Hz]
wn = 2*pi*f % cutoff [rad/s]
wn_stop = [0.8*f 1.2*f]*2*pi; %bandpass
% alipäästö
[b_low,a_low]=butter(n,wn,'low','s');
H_low = tf(b_low,a_low)
% ylipäästö
[b_high,a_high]=butter(n,wn,'high','s');
H_high = tf(b_high,a_high)
% kaistanesto
[b_stop,a_stop]=butter(n,wn_stop,'stop','s');
H_stop = tf(b_stop,a_stop)
% kaistanpäästö
[b_pass_low,a_pass_low]=butter(n,wn_stop(2),'low','s');
H_pass_low = tf(b_pass_low,a_pass_low);
[b_pass_high,a_pass_high]=butter(n,wn_stop(1),'high','s');
H_pass_high = tf(b_pass_high,a_pass_high)
H_pass = H_pass_low*H_pass_high;
% bode-kuvaajat
figure(1); clf;
bode(H_low,H_high,H_stop,H_pass);
legend('low','high','stop','pass');
```

Butterworth

```

% Butterworth
n = 5; % order
f = 0.5 % cutoff frequency
wn = 2*pi*f
wn_stop =
% alipäästö
[b_low,a_low] = butter(n,wn);
H_low = tf(b_low,a_low);
% ylipäästö
[b_high,a_high] = butter(n,wn);
H_high = tf(b_high,a_high);
% kaistanpoisto
[b_stop,a_stop] = butter(n,wn);
H_stop = tf(b_stop,a_stop);
% kaistanpäästö
[b_pass_low,a_pass_low] = butter(n,wn);
H_pass_low = tf(b_pass_low,a_pass_low);
[b_pass_high,a_pass_high] = butter(n,wn);
H_pass_high = tf(b_pass_high,a_pass_high);
H_pass = H_pass_low * H_pass_high;
% bode-kuva
figure(1);
bode(H_low,H_high,H_stop,H_pass);
legend('low','high','stop','pass');

```



Butterworth simulointikoodia 2

```

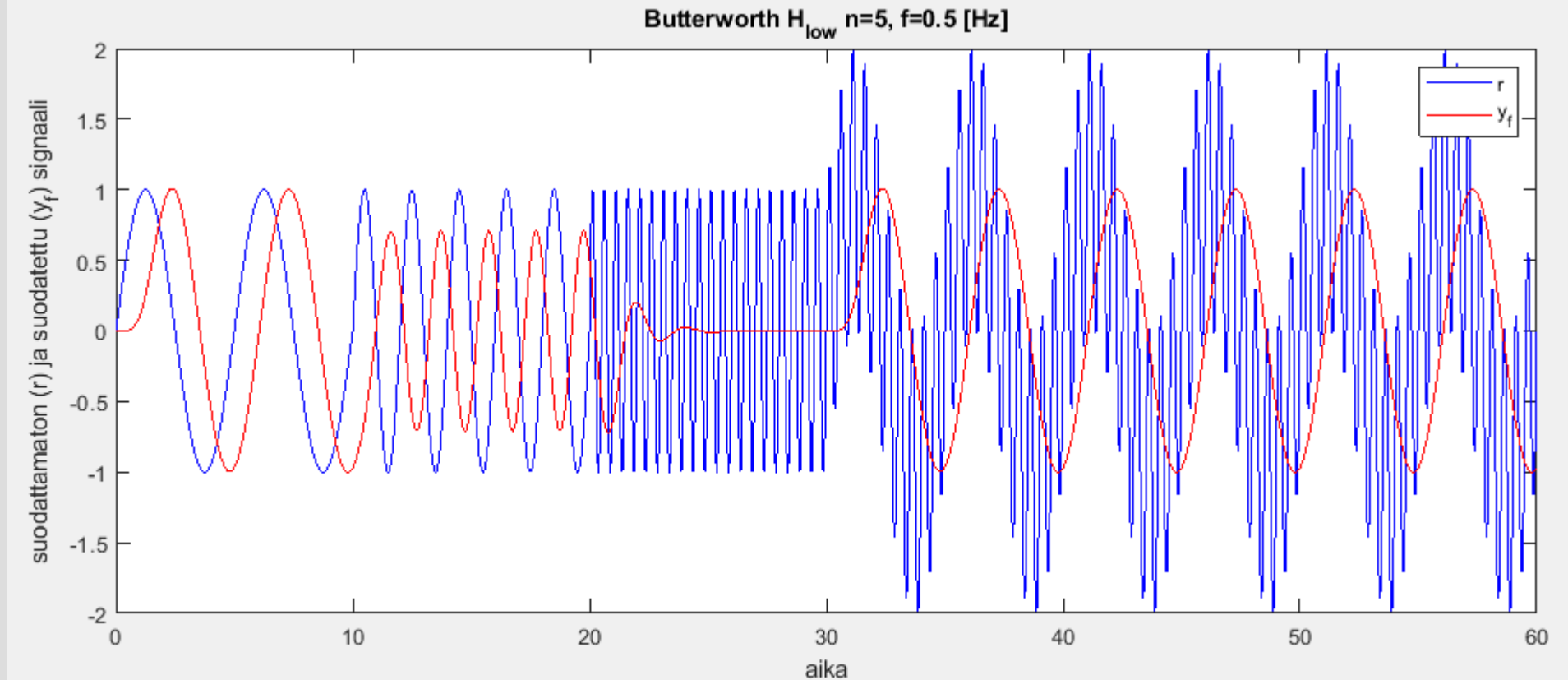
% aikataason simulointi
figure(2)
t=[0:0.001:60]'; % minuutin simulointi [s]
F=[0.2 0.5 2 ] % taajuus sisäänmenossa [Hz ]
for i=1:length(t)
    if t(i)<10, ff=F(1); % kun 0<t<10
    elseif t(i)<20, ff=F(2); %kun 10<=t<20
    elseif t(i)<30, ff=F(3); %kun 20<=t<30
    end
    r(i) = sin(2*pi*ff*t(i));
    if t(i)>=30 %kun 30<=t<60
        r(i) = sin(2*pi*F(1)*t(i)) +sin(2*pi*F(3)*t(i));
    end
end
end
yf = lsim(H_low,r,t);
%yf = lsim(H_pass,r,t);
%yf = lsim(H_stop,r,t);
plot(t,r,'b-',t,yf,'r-')
ylabel('suodattamaton (r) ja suodatettu (y_f) signaali');
xlabel('aika')
title(sprintf('Butterworth H_{low} n=%d, f=%1.3g [Hz]',n,f))
legend('r','y_f')

```

Butterworth simulointikoodia 2

% aikataso simulointi

figure(2)



xlabel('aika')

title(sprintf('Butterworth H_{low} n=%d, f=%1.3g [Hz]',n,f))

legend('r','y_f')

