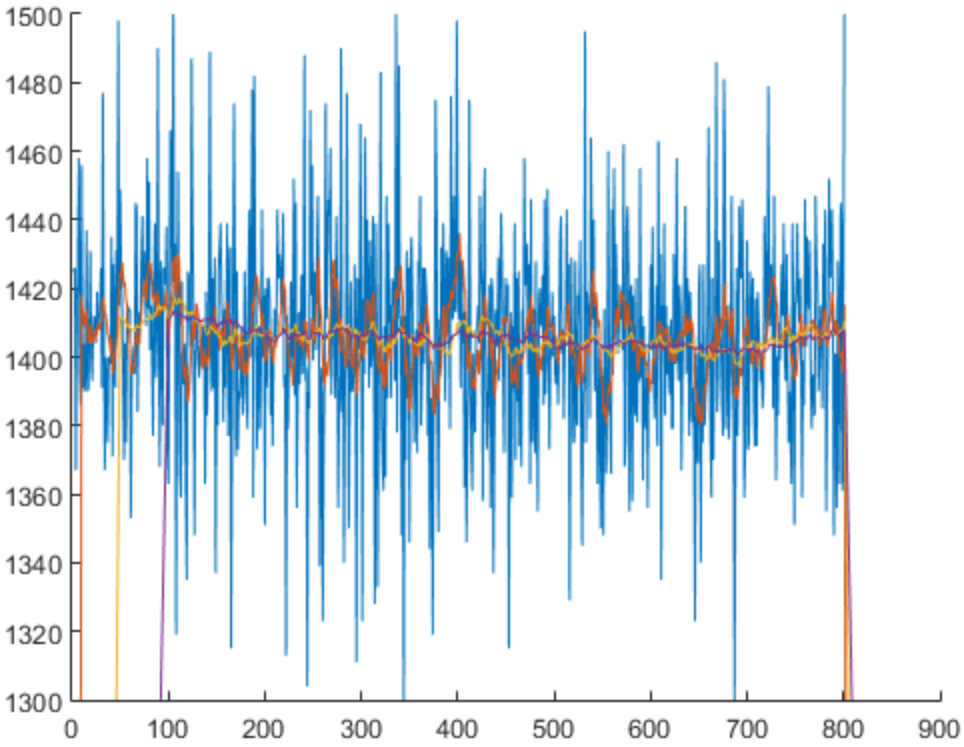
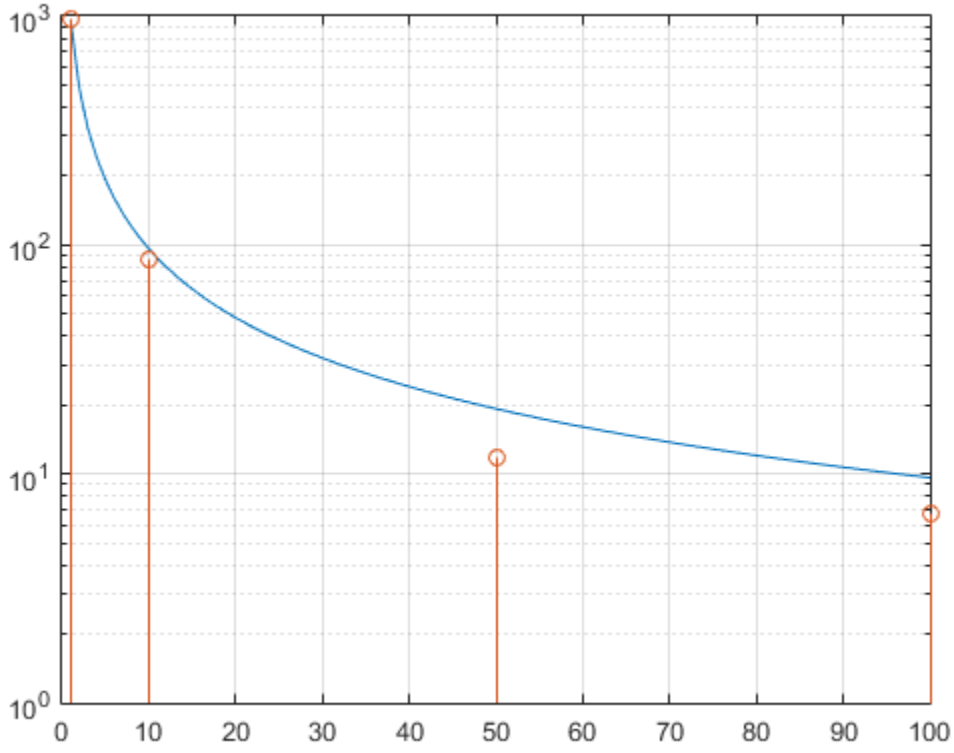
Table of Contents

..... 1
Krav max indsvingningstid 100ms 3
Step test for at se om det opfylder kravet. 4
Eksponentielt midlingsfilter. 5
ekstra opgave håndtering af korrupt data. med median filter 8


```
close all
clear
clc
load('vejecelle_data.mat'); % 1kg under 1020 -- uden lod over 1020
data=vejecelle_data;
clear vejecelle_data;
data_mlod = data(100:900);
data_ulod = data(1500:2300);


h10=ones(1,10)/10;
h50=ones(1,50)/50;
h100=ones(1,100)/100;

data_ulod_10=conv(h10,data_ulod);
data_ulod_50=conv(h50,data_ulod);
data_ulod_100=conv(h100,data_ulod);

figure
hold on
plot(data_ulod)
plot(data_ulod_10)
plot(data_ulod_50)
plot(data_ulod_100)
ylim([1300,1500])


avg_meas=[mean(data_ulod), mean(data_ulod_10(11:end-11)),
 mean(data_ulod_50(51:end-51)), mean(data_ulod_100(101:end-101))];
rms_meas=[std(data_ulod), std(data_ulod_10(11:end-11)),
 std(data_ulod_50(51:end-51)), std(data_ulod_100(101:end-101))];
var_meas=rms_meas.^2;

var_calc=var_meas(1)./[1:100];

figure
semilogy([1:100],var_calc)
hold on
grid on
stem([1 10 50 100], var_meas)
```

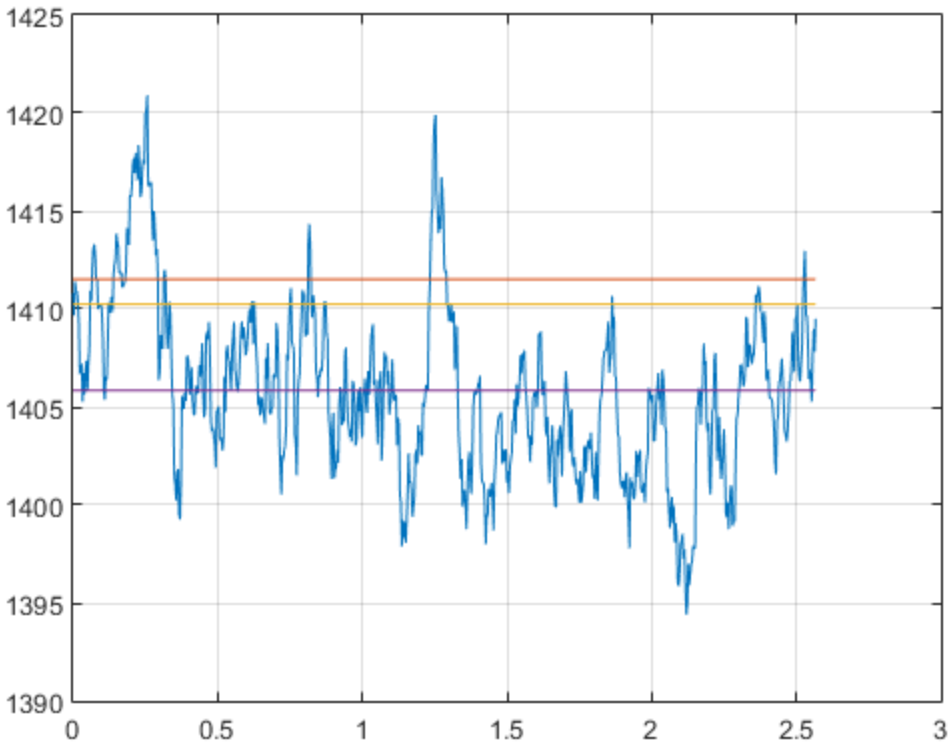

1500
1480
1460
1440
1420
1400
1380
1360
1340
1320
1300
0
100
200
300
400
500
600
700
800
900


10^3
10^2
10^1
10^0
0
10
20
30
40
50
60
70
80
90
100

Krav max indsvingningstid 100ms

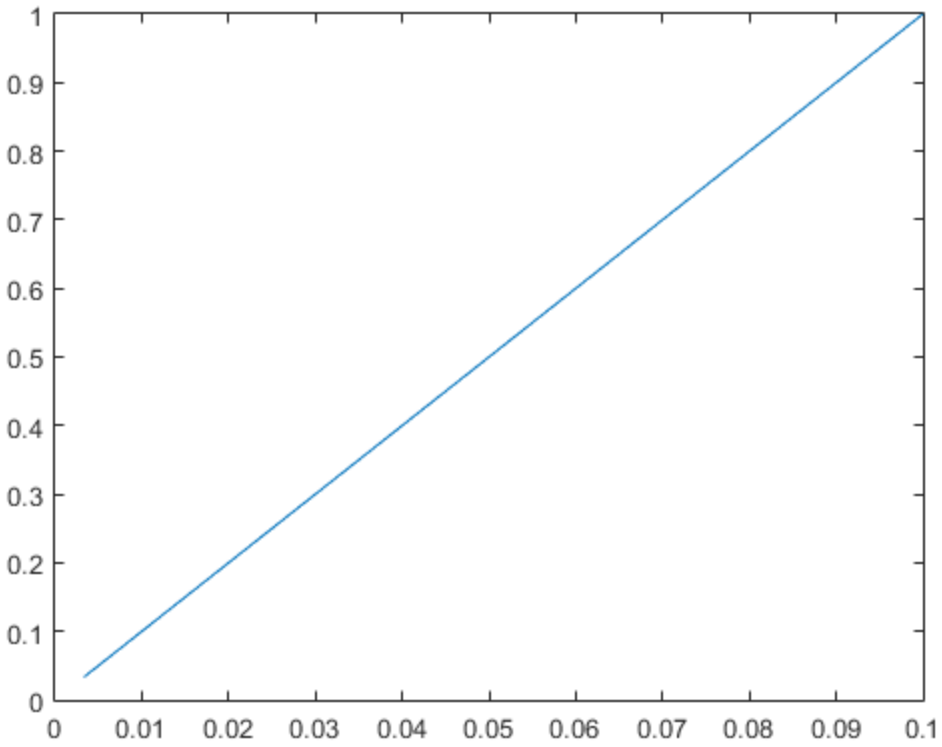
```
Ts=fs^-1

M=floor(0.1/Ts) % runder ned da ellers vil kravet ikke være opfyldt,
 dog
                % går tallet lige op så det er ikke noget problem men
 for
                % god ordens skyld bruges floor alligevel.

h_M=ones(1,M)/M;

data_ulod_M=conv(data_ulod,h_M);

mean_M=mean(data_ulod_M(M:end-M))

var_calc_M=var_meas(1)/M
rms_calc_M=sqrt(var_calc_M)

rms_M=std(data_ulod_M(M:end-M))
var_M=rms_M^2


figure
plot([0:length(data_ulod_M)-M*2]*Ts, data_ulod_M(M:end-M))
hold on
grid on
plot([0:length(data_ulod_M)-M*2]*Ts, (rms_calc_M
+mean_M)*ones(1,length(data_ulod_M)-M*2+1));
plot([0:length(data_ulod_M)-M*2]*Ts, (rms_M
+mean_M)*ones(1,length(data_ulod_M)-M*2+1));
plot([0:length(data_ulod_M)-M*2]*Ts,
 (mean_M)*ones(1,length(data_ulod_M)-M*2+1));


Ts =

    0.0033


M =

    30


mean_M =

   1.4058e+03


var_calc_M =

   32.0997
```

```
rms_calc_M =

    5.6657


rms_M =

    4.4001


var_M =

   19.3608
```



Step test for at se om det opfylder kravet.

```
step_M=conv(h_M, ones(1, M));

figure
plot([1:length(step_M)-M+1]*Ts,step_M(1:end-M+1)) % kravet er opfyldt.
```

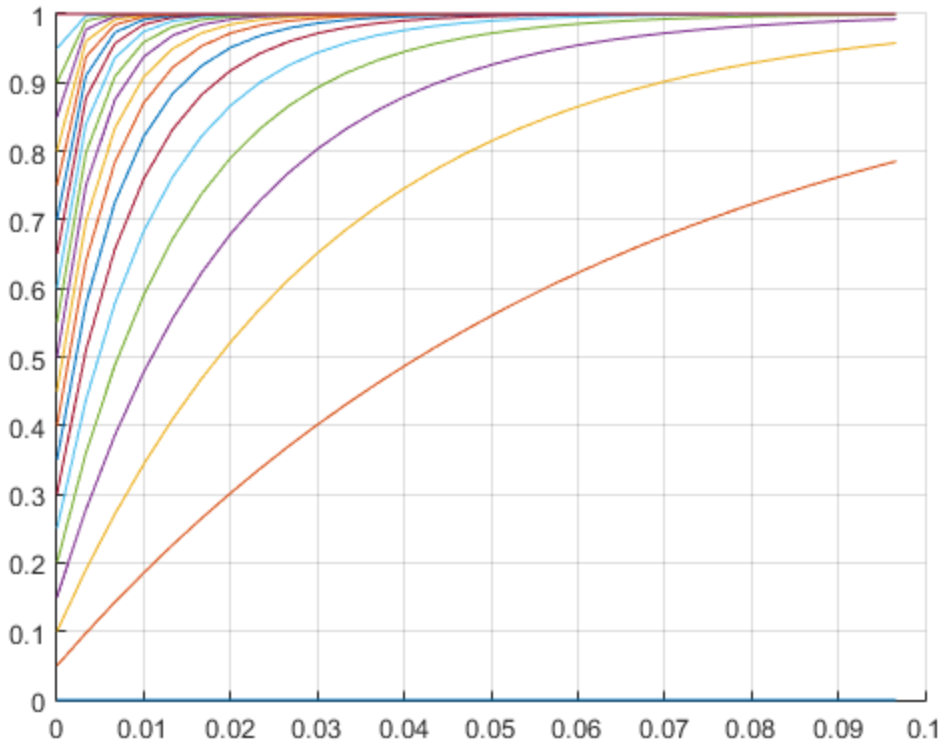


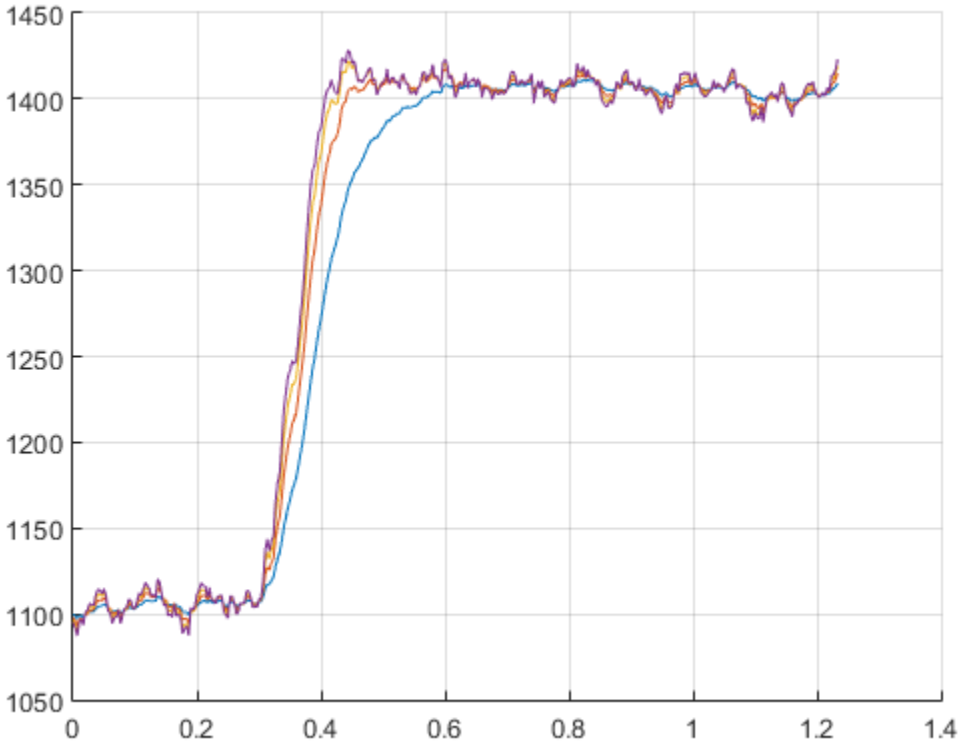
Eksponentielt midlingsfilter.

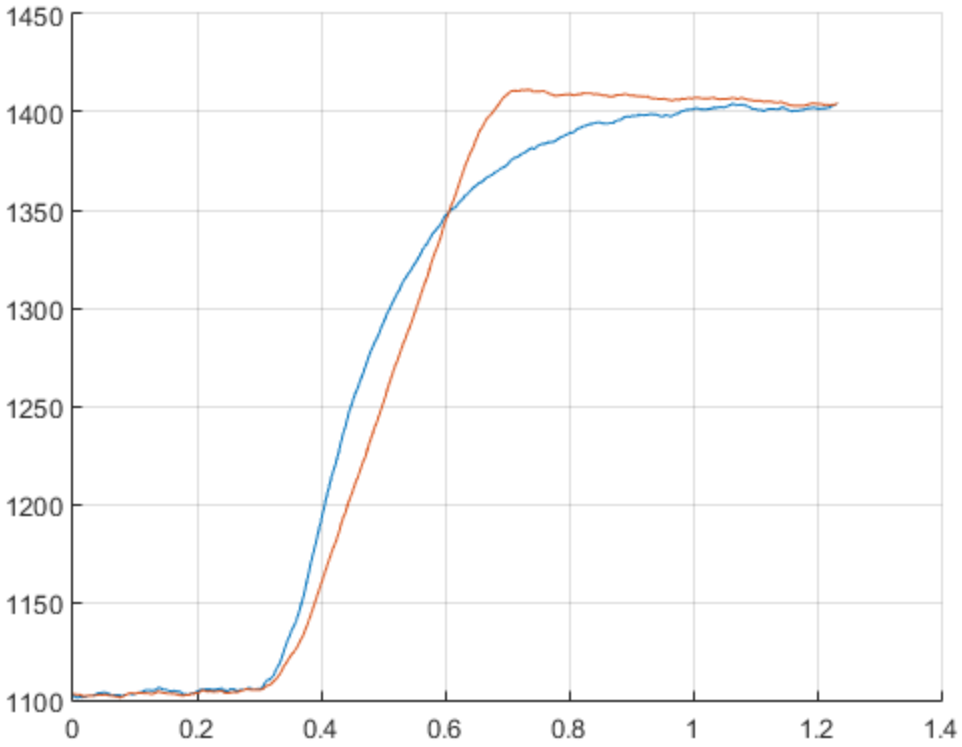
```
% test værdier af alfa.

x=ones(1,M);
figure
hold on
grid on
for a=[0:0.05:1]
   [b, a]=ExpMean(a);
   y=filter(b,a, x);
   plot([0:length(y)-1]*Ts, y);

end

x=data;
indsving=[930:1300];
figure
hold on
grid on
for a=[0.05:0.05:0.2]
   [b, a]=ExpMean(a);
   y=filter(b,a, x);
   plot([0:length(y(indsving))-1]*Ts, y(indsving));
end
```

```
% udfra dette kan det ses at alfa nok skal være omkrin 0.2.

% udregn hvad alfa skal være hvis den skal have samme dæmpning som
 vores
% 100 ordens filter. ay^2=(a/(2-a))*ax^2

a=(2*var_calc(100))/(var_meas(1)+var_calc(100))
[b, a]=ExpMean(a);
figure
hold on
grid on
y=filter(b,a, x);
plot([0:length(y(indsving))-1]*Ts, y(indsving));

rms_exp=std(y(1500:2300))
var_exp=rms_exp^2

var_calc(100)
rms_calc_100=sqrt(var_calc(100))

data_100=conv(h100,data);
plot([0:length(data_100(indsving))-1]*Ts, data_100(indsving));

% det kan ses at Eksponentielt midlingsfilter reagere hurtigere end
 det 100
% ordens filter, dog har den en længere indsvingningstid.
```

```
a =

    0.0198


rms_exp =

    3.2040


var_exp =

   10.2657


ans =

    9.6299


rms_calc_100 =

    3.1032
```


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1


1450
1400
1350
1300
1250
1200
1150
1100
1050
0
0.2
0.4
0.6
0.8
1
1.2
1.4

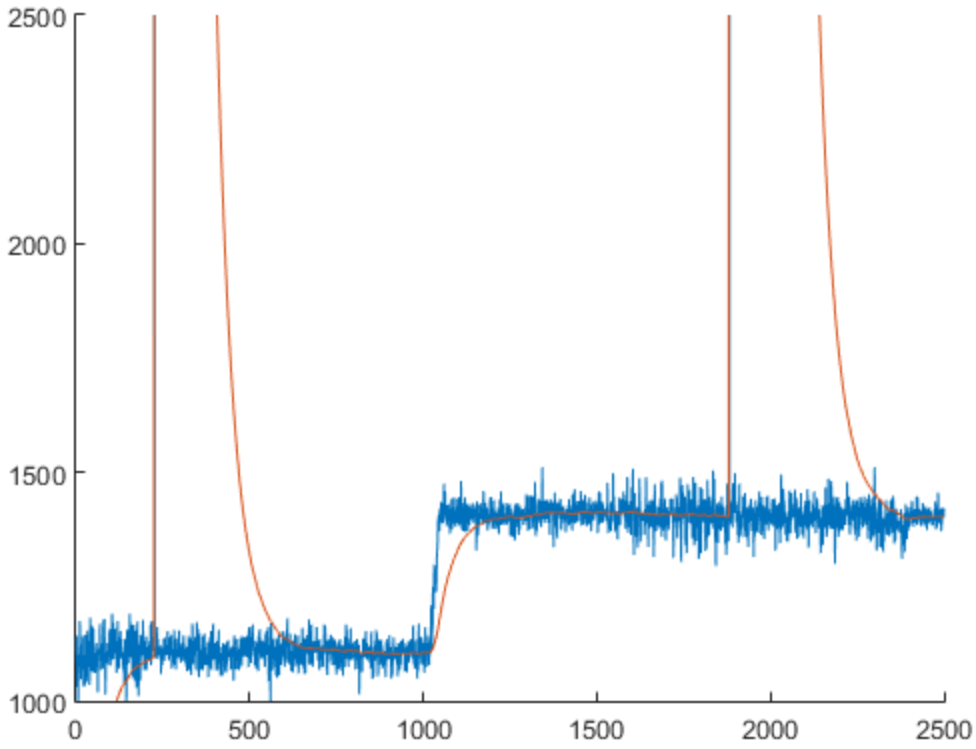


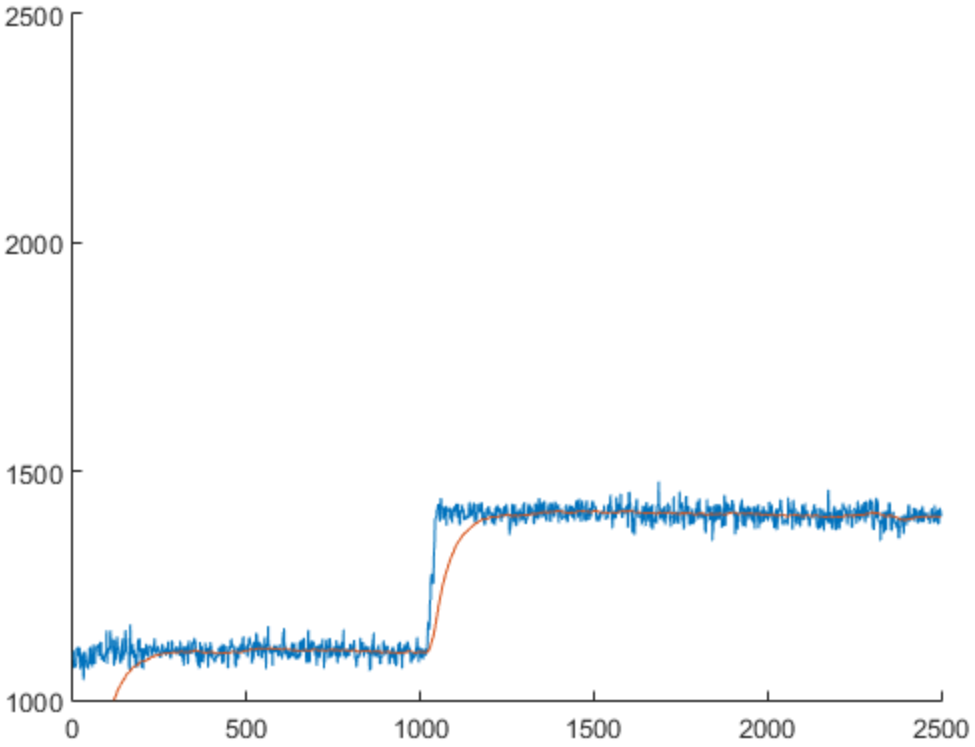
ekstra opgave håndtering af korrupt data. med median filter

```
data_korrupt=data;

for i=1:length(data_korrupt);

    if(rand >= 0.999)
        data_korrupt(i)=round((2^24)*rand);
    end
end


figure
hold on
plot(data_korrupt)
plot(filter(b,a,data_korrupt))
ylim([1000,2500])

y=medfilt1(data_korrupt, 3);
figure
hold on
plot(y)
plot(filter(b,a,y))
```

```
ylim([1000,2500])

eksempel=median([1 2 2 2 inf 1 1 1])
```

```
eksempel =

    1.5000
```





Published with MATLAB® R2015a