Aarhus universitet

Introduktion til digital signalbehandling

3. Semester

# DSA Case 2


| *Gruppemedlemmer:* | *AUid:* |
|---|---|
| Gustav A. Gammelgaard | au538293 |
| Stinus Lykke Skovgaard | au520659 |
| Tim Hede Stenholt | au543518 |

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING


31. marts 2017

# Indhold

# Figurer

# 1 Problem beskrivelse

Denne case omhandler måden at forbedre et støjfyldt signal med et midlingsfilter. Dette filter skal midle et signal fra en vægt. Typisk vil signalet fra vægten svinge voldsomt som vil gøre det svært at måle korrekt. Midlingsfilteret vil gøre det muligt at få en pålidelig måling. Filteret er realiseret ved hjælp af matlab og vægt data er importeret ind i matlab og testet i dette miljø.

# 2 Opgave 1

# 3 Opgave 2

Det kommende afsnit kommer til at omhandle hvordan midlingsfilteret virker og hvordan det er implementeret. Filteret er blevet testet på dataen fra opgave 1. Vi har ekspermenteret med både lineært og eksponentielt midlingsfilter.

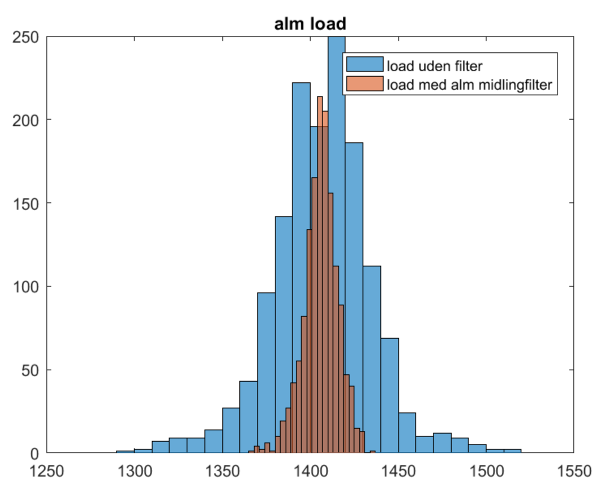
## 3.1 Lineært midlingsfilter

For at midle data fra opgave 1, lavede vi først et lineært midlingsfilter. Dette er gjort på følgende måde:

```
M = 10;
h1 = 1/M * ones(1, M);
```

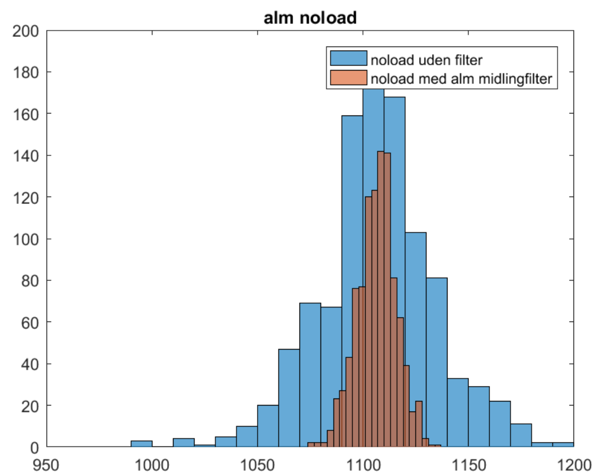
Dette laver et midlingsfilter med M koeficienter, der hver har en værdi på 1/M. Ved at indstille på ordnen(altså M) vil man kunne få filteret til midle kraftigere.

Da dataen fra opgave 1 viser at der både er en load og noload, er filteret påført de to stadier hver for sig. Dette er blevet plottet på et histogram, som kan ses på figur 1



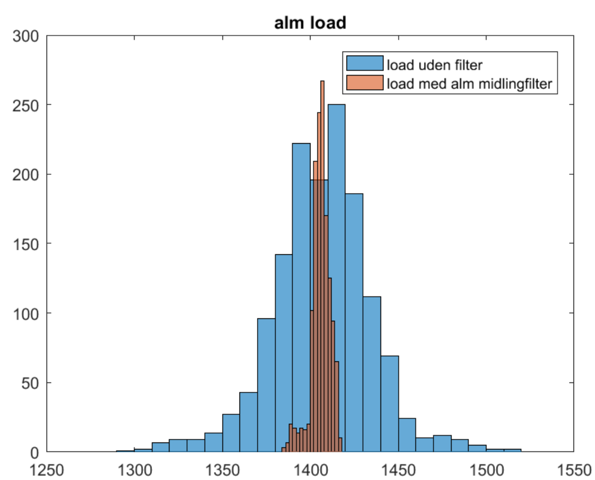

Figur 1: Histogram af load med og uden filter. Filterorden = 10

Man kan se at filteret får vores data til at ligge tættere på hinanden, hvilket er det vi gerne vil se. Det samme kan ses for noload på 2




Figur 2: Histogram af noload med og uden filter. Filterorden = 10

Ved at ændre på ordnen kan man få filtret til at midle kraftigere. Dette kan ses på figur 3




Figur 3: Histogram af load med og uden filter. Filterorden = 50

Det ses tydligt at de orange pinde er blevet smallere og fylder et mindre område på grafen.
Hvis man kigger på varians og spredning kan man se at de stemmer godt overens med vores grafer.

```

var_load = var(y_load)
var_noload = var(y_noload)

P_load = (1/sqrt(M)*S_load)^2
P_noload = (1/sqrt(M)*S_noload)^2
```

Ved M = 10 får vi en varians på ca 90, mod en varians på ca 25 ved en orden på 50. Hvis man kigger på støj-effekten ser man det samme. Ved en orden på 10 er støj-effekten ca 10 højere end ved en orden på 50.

Hvis man ønsker en max svingningstid for sit system, bliver man muligvis nødt til at begrænse sin orden på filteret. Hvis vi har et krav om en indsvingningstid på 100ms kan vi lave en hurtigt udregning:

```
maxKoef = fs*0.1;
```

Dette giver os et filter med en orden på 30.

## 3.2 Eksponentielt midlingsfilter

Vi forsøgte også at lege med et eksponentielt midlingsfilter. Dette midlingsfilter bruger et simpelt IIR lavpasfilter beskrevet af differensligningen:

$$y(n) = \alpha \cdot x(n) + (1 - \alpha) \cdot y(n - 1)$$

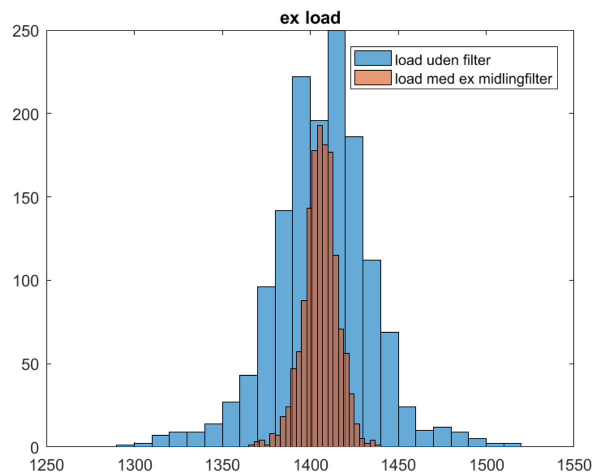
Denne type filter er ofte brugt pga dens fordele, som fx at den kræver færre beregninger per output-sample end et standard midlingsfilter, og dens stærkt reduceret hukommelses brug, idet den kun indeholder et forsinkelse element $y(n-1)$, som skal gemmes i en hukommelses-plads i fx. et embedded system. De nyeste samples $x(n)$ får også størst betydning i outputtet $y(n)$, hvilket betyder at filteret reagerer hurtigere på ændringer i input $x(n)$ et standard midlingsfilter.
$\alpha$ er en midlingsfaktor mellem 0 og 1 som bestemmer støjreduktionen i forhold til responstiden. Vi prøvede med en lille værdi af $\alpha$ i vores midlingsfilter, hvilket gav en god støjreduktionen men langsom responstid, omvendt prøvede vi også med en høj værdi af $\alpha$ i vores midlingsfilter, hvilket gav en mindre støjreduktionen, men hurtigere responstid. Så præcis som med det alm midlingsfilter er der et trade off mellem støjreduktionen og responstid.
$\alpha$ bestemmes ved:

$$\alpha = \frac{2}{R + 1}$$

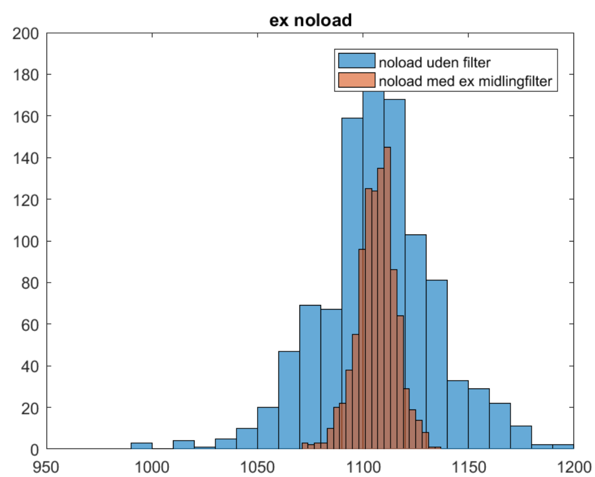
Hvor R er den faktor som støj variensen bliver reduceret. Vi kan sammenligne den eksponentiele midlingsfilters reduktion i støjeffekten direkte med et linært midlingfilters reduktion ved at opskrive:

$$\alpha = \frac{2}{N + 1}$$

Hvor N er antal af FIR midlings koeficienter, når $N > 3$. Dette har vi prøvet at vise midligning for et eksponentielt midlingsfilter med $N = 10$ stemmer nogenlunde med midlingen fra et $N = 10$ point linært midlingsfilter fra før. De nye midlinger eksponentielt midlingsfiltere kan ses i Figure 4 og Figure 5

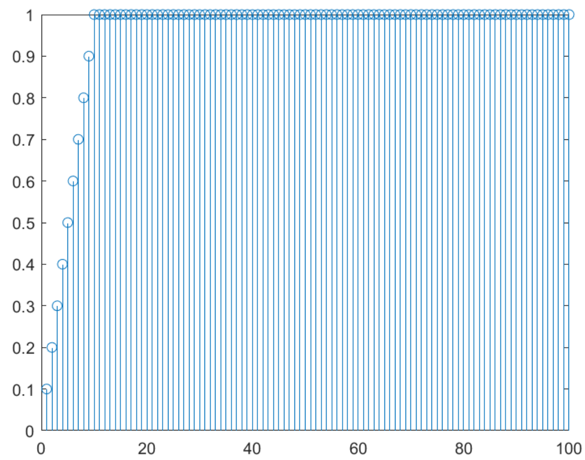



Figur 4: Histogram af load med og uden eksponentielt filter. N = 10

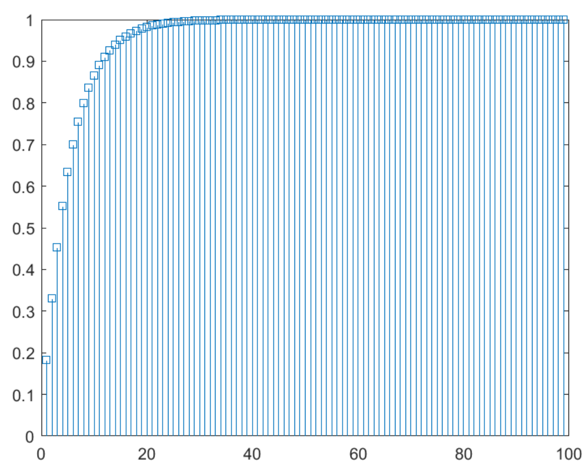



Figur 5: Histogram af noload med og uden eksponentielt filter. N = 10

Som man kan se minder midlingen for et eksponentielt midlingsfilter i Figure 4 og Figure 5 med med midlingen fra de linære midlingsfilter fra før i Figure 1 og Figure 2
Den store forskel er deres respons på fx. et step som kan ses i Figure 6 og Figure 7




Figur 6: Linært midlingsfilter respons på step ved M=10.




Figur 7: Ekspontielt midlingsfilter respons på step ved N=10.

Som det kan ses i Figure 6 og Figure 7, Så er reagerer det eksponentielle midlingsfilter hurtigere på steppet end det linære midlingsfilter, dog indhentes forspringet fra det eksponentielle midlingsfilter, idet det aldrig kan ramme slutværdien præcis, men blot komme uendelig tæt på. Alligevel tager det ca dobbelt så mange samples for det eksponentielle midlingsfilter at nå til næsten slutværdiden i forhold til det linære filter som når slutværdien efter dens 10 kofficenter er kørt igemme signalet. Dette skyldes at eksponentielle midlingsfilter er et IIR filter som jo har uendeligt respons på et siganal.

# 4 Opgave 3

# 5 Matlab kode