

AARHUS UNIVERSITET

DSB

SEMESTER 3

---

## Mini-projekt - Del 3

---

*Studerende:*

Mette HAMMER NIELSEN-KUDSK - STUDIENR: 201408391

Martin BANASIK - STUDIENR: 201408398

Finja JETTE RALFS - STUDIENR: 201303659

November 15, 2015



# Indholdsfortegnelse

1	FIR - Finite impulse response	2
2	IIR - Infinite impulse response	2
3	Z-transformation	3
4	Foldning	4
5	Impulsfunktion: $\delta(n)$	4
6	Vindues metoden	4
7	Filter design i MatLab	4
8	Filtrering af EKG signal	5
9	Fremtidigt arbejde	10
10	Konklusion	10

# 1 FIR - Finite impulse response

En vægtet sum (udglatning, midling), som kører hen over input-signalet  $x(n)$ .

Differensligning for FIR-filteret og definition:

$$y(n) = \sum_{k=0}^{M-1} b_k * x(n - k)$$

hvor,

$$x(n) = \text{input}$$

$$y(n) = \text{output}$$

$$b_k = \text{filteretskoefficienter}$$

$$M = \text{filteretslængde}$$

$$M - 1 = \text{filteretsorden}$$

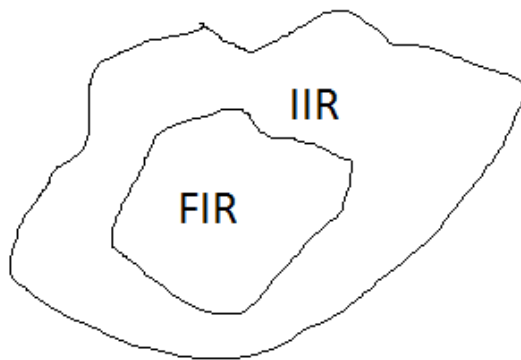


Figure 1: FIR filter hører under IIR.

Ved brugen af et FIR filter stopper outputtet.

# 2 IIR - Infinite impulse response

IIR er en anden type for et digitalt filter, der kan have et uendeligt respons på en impuls.

Differensligning for FIR-filteret og definition:

$$y(n) = \sum_{k=0}^M b_k * x(n - k) - \sum_{k=1}^N a_k * y(n - k)$$

hvor,

$$x(n) = \text{input}$$

$$y(n) = \text{output}$$

$$b_k = \text{filteretsfeedforwardkoefficienter}$$

$$M = \text{filteretsorden}$$

$$a_k = \text{filteretsfeedbackkoefficienter}$$

Ved brugen af et IIR filter stopper outputtet aldrig.

### 3 Z-transformation

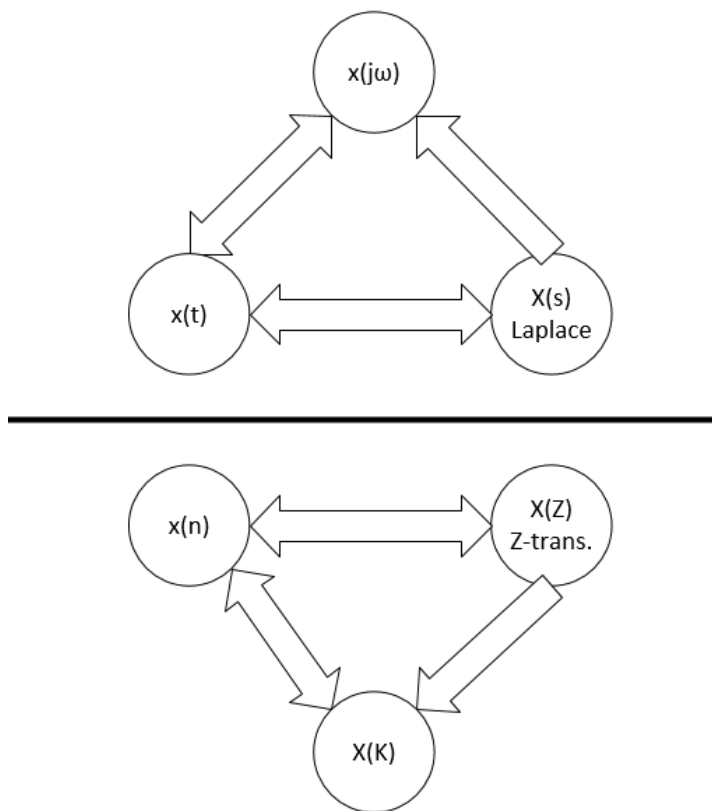


Figure 2: Transformationer.

Når vi snakker IIR filtre, så kan disse ofte være meget forvirrende, og det kan derfor være svært at se om filteret er stabilt eller ustabil. Her indføres derfor z-transformation. Definition:

$$X(z) = \sum_{n=0}^M x(n) * z^{-1}$$

Der er nogle forskellige ting, der er vigtige at vide ved z-transformation:

$$\delta(n)$$

$$z\{\delta(n)\} = 1$$

Hvad ville der sker hvis impulsen er delayed?

$$\delta(n-1)$$

$$z\{\delta(n-1)\} = z^{-1}$$

$z^{-1}$  kaldes for unit delay.

Vi kender, fra Analoge signal, til overføringsfunktioner. Det samme gælder her:

$$H(z) = \frac{\text{output}}{\text{input}} = \frac{Y(z)}{X(z)}$$

Når vi har fået lavet vores z-transformation kan vi altså finde overføringsfunktionen for filteret. Igennem overføringsfunktionen kan vi, i tælleren find ud af om filteret har nogle nulpunkter og i nævneren finde polerne. Hvis IIR filteret har 1 pol er det et 1. ordens IIR filter, hvis der havde været 2 poler, havde filteret været et 2. ordens IIR filter. Det er altså derfor nævneren, i overføringsfunktionen, hvor man kan finde polerne, der er interessant. Man kan herudfra se om filteret er stabilt eller ej. Hvis polerne er under eller lig 1 er filteret stabilt, hvis polerne er over 1 er filteret ustabil. Da FIR filter altid er stabile, giver det ingen mening at lave z-transformation på disse.

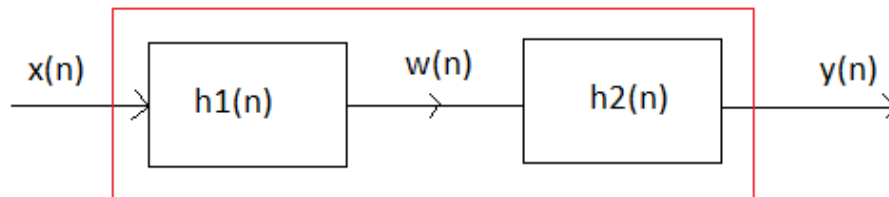
## 4 Foldning

Hvis vi ønsker at koble flere filter sammen, benytter vi kobling.

Definition:

$$y(n) = \sum_{k=0}^{M-1} h(k) * x(n - k)$$

I MatLab benytter vi, ved foldning, funktionen "conv". Hvis vi har to signaler vi vil have koblet sammen, så repræsenteres dette ved en stjerne:



Den røde firkant omkredser de to signaler, der bliver foldet sammen.

$$h_1(n) \star h_2(n)$$

## 5 Impulsfunktion: $\delta(n)$

## 6 Vindues metoden

Når vi analytisk skal designe et filter, benytter vi os af vinduesmetoden. Her befinder vi os i frekvensverden (DFT). 1) Vi specificer det ønskede frekvens karakteristisk. 2) Herefter laver vi Invers DFT (IDFT). 3) Så ganger vi vinduet på. 4) Til sidst tjekker vi frekvenskarakteristik med DFT.

## 7 Filter design i MatLab

Vi benytter os af to forskellige funktioner "fir1" og "fir2".

- "fir1":

$$b = \text{fir1}(N, \frac{f_c}{0.5 * f_s}, 'high')$$

hvor,

$$f_c = \text{knækfrekvensen}$$

$$f_s = \text{samplesfrekvens}$$

Her vil MatLab lave et højpas filter, da vi har indskrevet 'high'. Hvis vi ønskede et lavpas filter skulle vi ikke havde skrevet noget. Ønskede vi et båndpas filter, ville vi skulle skrive bandpas og ved båndstop filer, bandstop.

- "fir2":

$$b = \text{fir2}(N, \dots, \dots)$$

Her skal der indsættes amplituder. Benyt "help" funktionen i MatLab foran "fir2", hvis der ønskes mere information om denne funktion.

Vi har i løbet af skriveprocessen lært at benytte funktionen "fdatool". Den er blevet brugt i løbet af dette miniprojekt til at designe et FIR lavpas filter og et FIR båndstop filter.

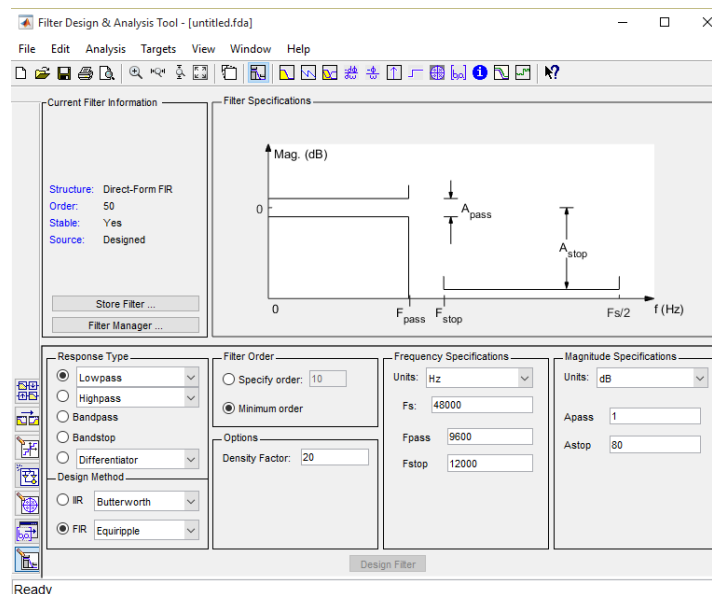


Figure 3: fdatool i MatLab

## 8 Filtrering af EKG signal

Der er blevet fundet et støjfyldt EKG-signal inde på siden Physionet. Her er EKG-signalet vist og som man kan se er der utrolig meget støj lige ved de 50 Hz. Det kan ses i tidsdomænet at EKG-signalet er ret støjfyldt efter hvert pulsslag, mellem PR-segmenterne og QT-segmenterne. Det ønsker vi fjernet.

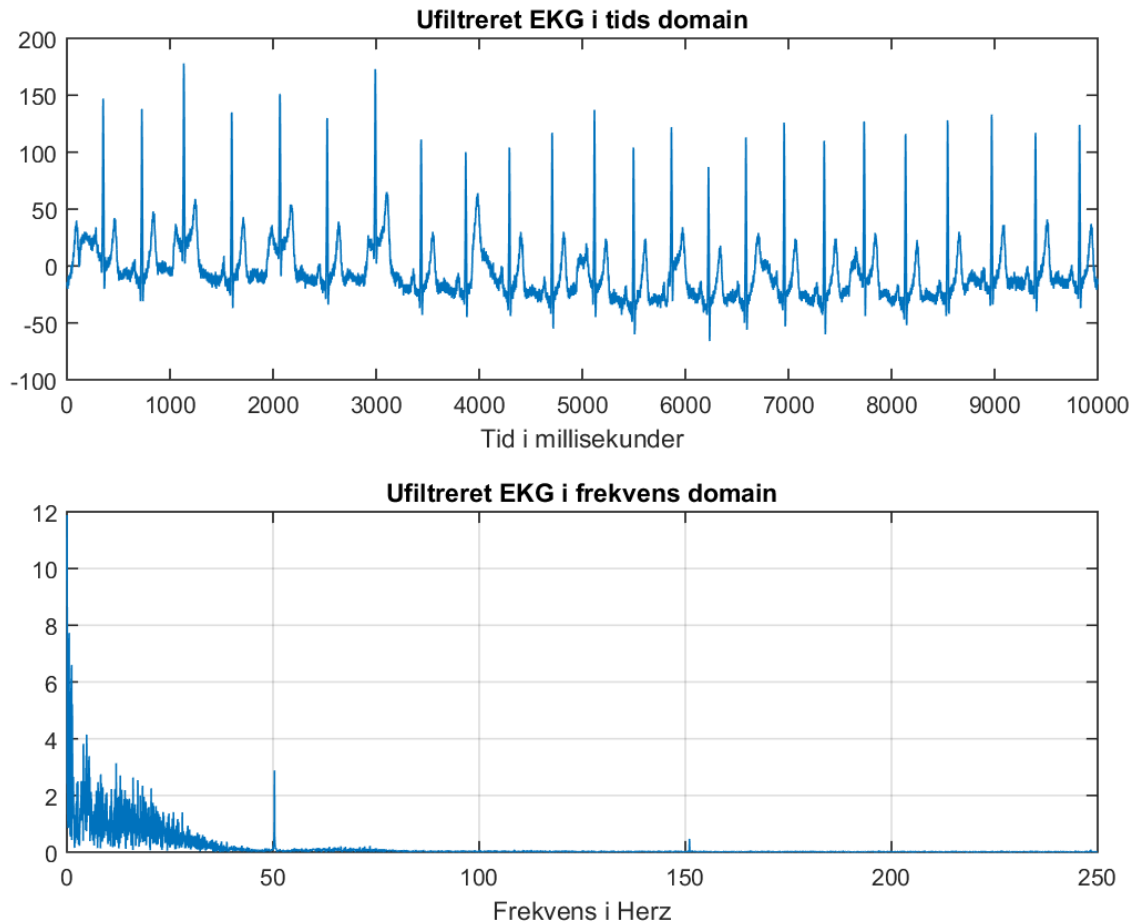


Figure 4: EKG ufiltreret.

Det mest iøjnefaldende er støjen ved de 50 Hz, så dem starter vi med at få fjernet. Det gøres i form af et IIR båndstopfilter. Det bliver designet vha. af MatLab funktionen "fdatool". Her indtastes:

$$Fs = 500$$

$$F_{pass1} = 47$$

$$F_{stop1} = 48$$

$$F_{stop2} = 52$$

$$F_{pass2} = 53$$

Vi har altså ønsket en knækfrekvens på de 50 Hz. Filteret loades herefter ind i MatLab filen og ligges ovenpå det støjfyldte EKG-signal i frekvensdomænet. Det kan nu ses på figurerne herunder at støjen ved de 50 Hz er blevet fjernet. Hvis vi sammenligner figur 3 og 4 i tidsdomænet kan det ses at støjen mellem de forskellige segmenter er mindsket. Det samme gælder i frekvensdomænet, hvor der i figur 3, kan ses støjen ved de 50 Hz - den er væk i figur 4 efter vi har kørt EKG-signalet igennem båndstop filteret. Hvis man kigger godt efter på frekvensdomænet i figur 4, kan man se lidt støj mellem 35 og 85 Hz. Det ønsker vi også fjernet.

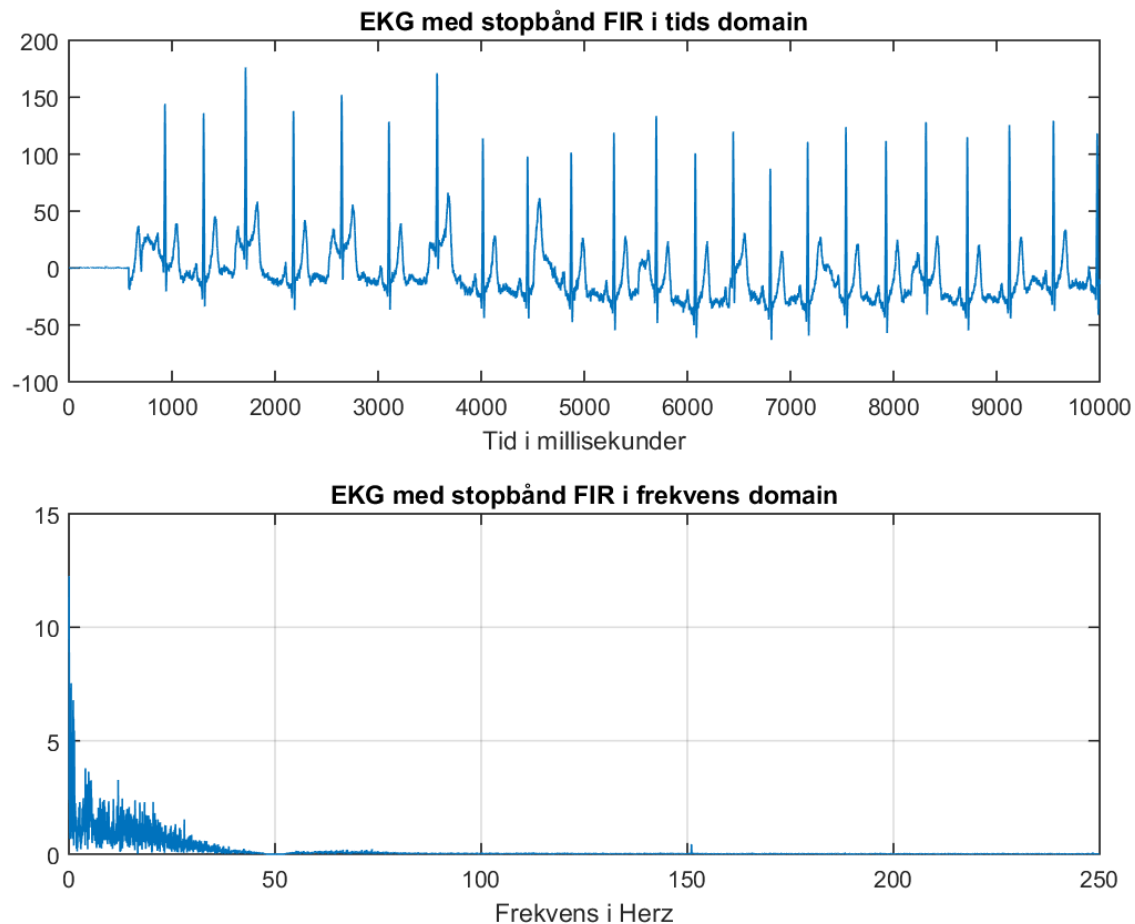


Figure 5: EKG med båndstop FIR.

Dette fjernes med et FIR lavpasfilter, der også bliver designet vha. "fdatool". Her indtastes:

$$F_s = 500$$

$$F_{pass} = 35$$

$$F_{stop} = 40$$

Vi har altså ønsket en knæfrekvens på 38 Hz. Hvis man kigger på figur 5 og sammenligner med figur 4, kan man nu se i frekvensdomænet, at støjen efter de 35 Hz er blevet fjernet. Igen kan vi se i tidsdomænet at vores EKG-signal er blevet meget pænere og mindre støjfyldt. Hvis vi går helt tilbage og sammenligner figur 3 og 5, kan vi se at EKG-signalet er meget pænere.



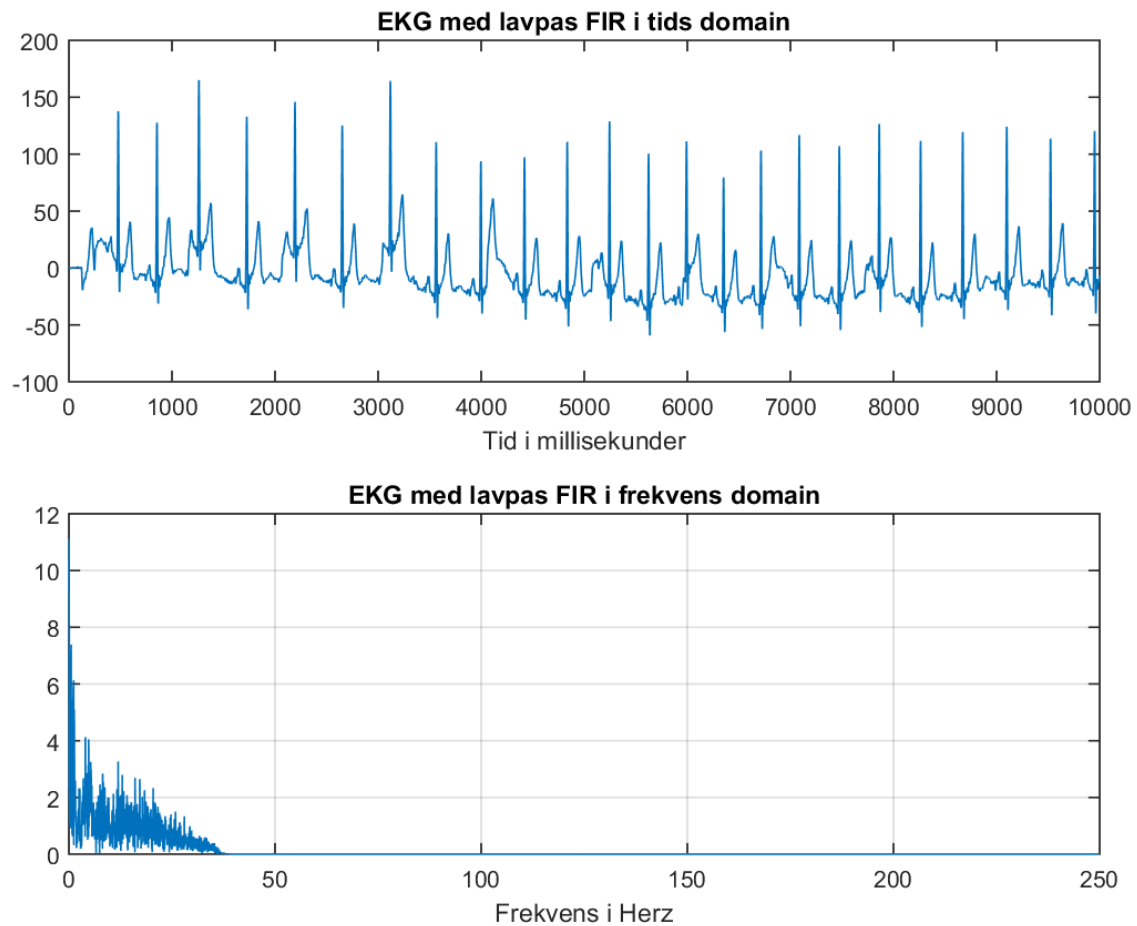


Figure 6: EKG med lavpas FIR.

For lettere at se forbedringerne har vi sat forbedringerne under hinanden vha. funktionen "subplot". Her har vi først EKG-signalet i tidsdomænet, hvor man tydeligt kan se at støjen er blevet fjernet.

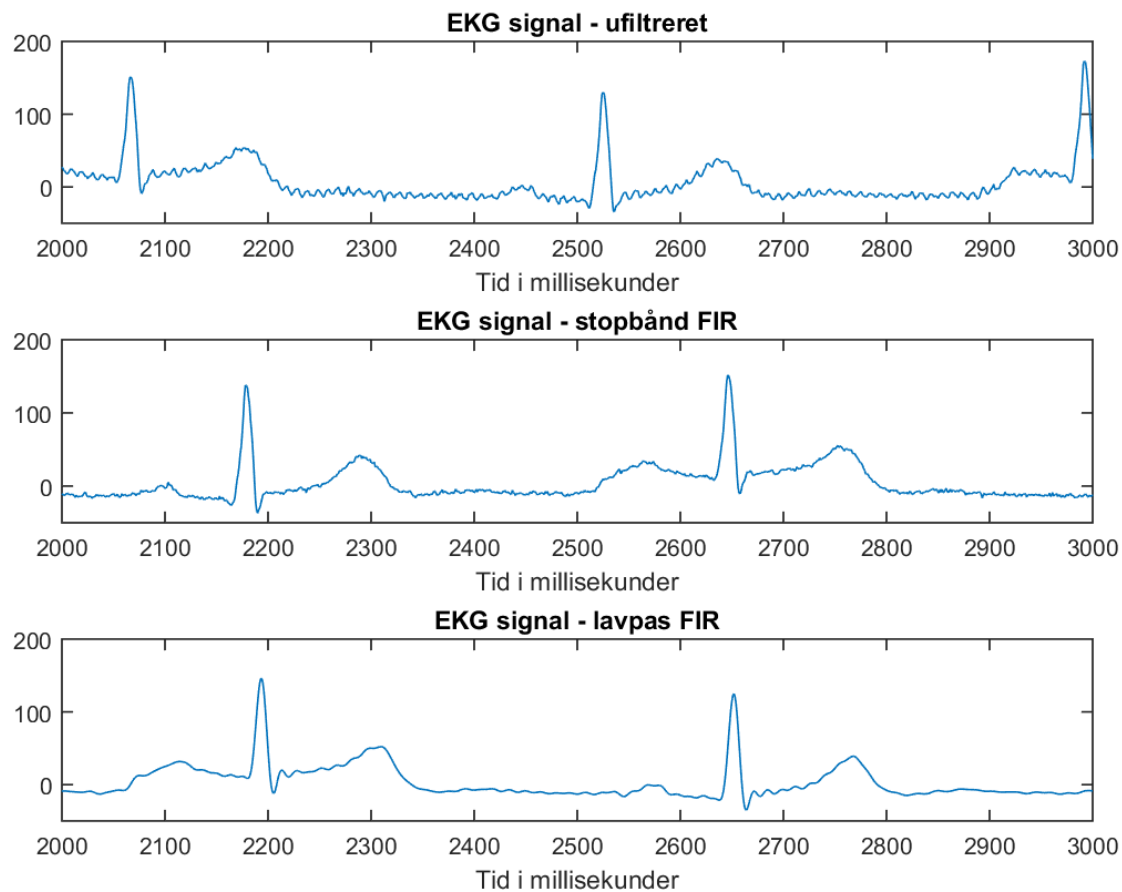


Figure 7: Sammenligning af EKG i tidsdomæne.

Her har vi sammenligningen af signalerne i frekvensdomænet. Ved det ufiltreret EKG-signal ser vi tydeligt støjen ved de 50 Hz. Dem fik vi fjernet vha. båndstop filteret. Hvis man kigger godt efter kan man faktisk også se lidt støj ved 150 Hz. Da vi ønskede at fjerne støj fra 35 Hz og frem, så fik vi også fjernet støjen ved de 150 Hz.

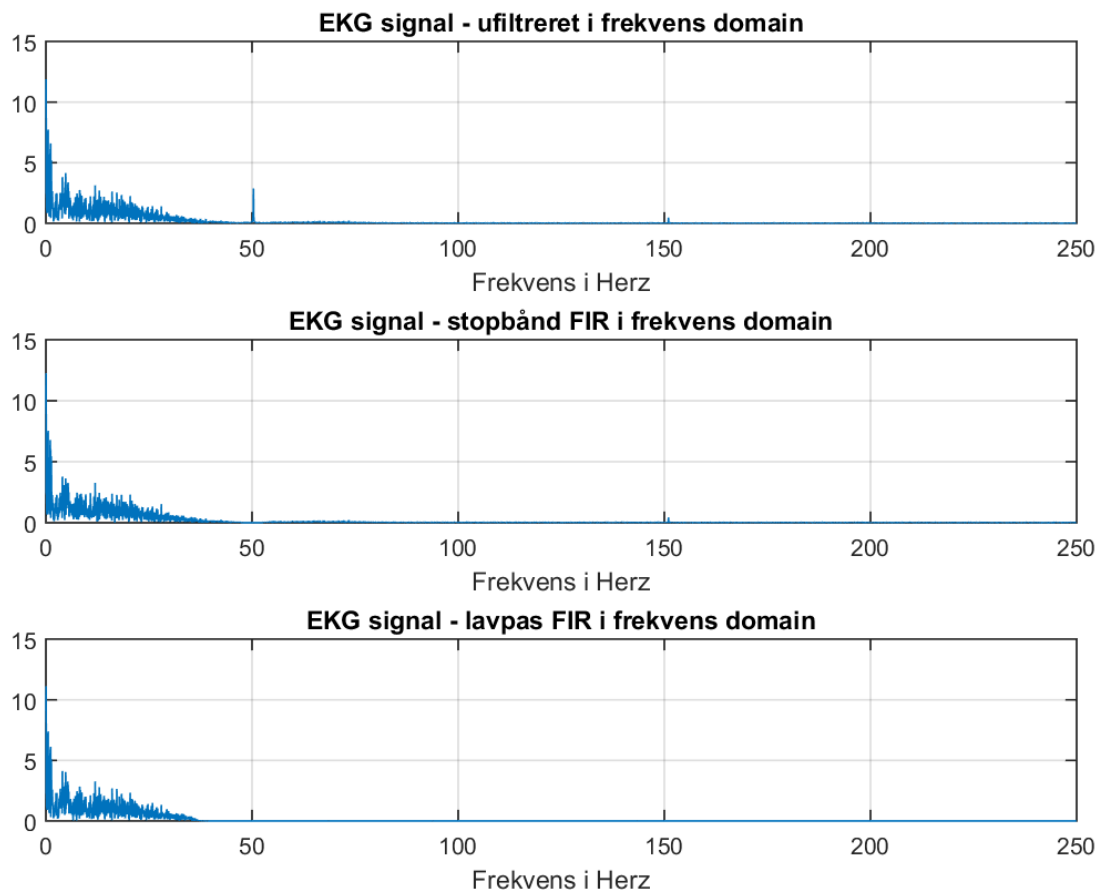


Figure 8: Sammenligning af EKG i frekvensdomænet.

## 9 Fremtidigt arbejde

Vi ønsker at få lidt bedre styr på FIR og IIR filtre og hvordan de helt præcis hænger sammen. Vi har fået en god forståelse af hvad de gør, men vil alle gerne vide mere. Dette undersøges til næste miniprojekt og er derved vores fremtidige arbejde.

## 10 Konklusion

Vi kan konkludere at funktionen "fdatoool" er super lækker at benytte når man skal have designet et filter. Man slipper for at bøvle med at skrive selve koden og hvordan MatLab funktionerne som f.eks. "fir1 og fir2" skal opbygges. Selvfølgelig skal man stadig kende fs og fc osv. men det giver jo sig selv. Vi har fået et klart større indblik i hvad de forskellige filtre kan, også ift. IIR og FIR filtre. Det giver meget bedre mening nu, efter man har siddet og leget med det og forsøgt sig frem med de forskellige filtre. Man har kunne se hvad virker på hvad og hvad helt sikkert ikke fungerer.