

# Kursusopgave 1

Louise Wichmann Moesgaard

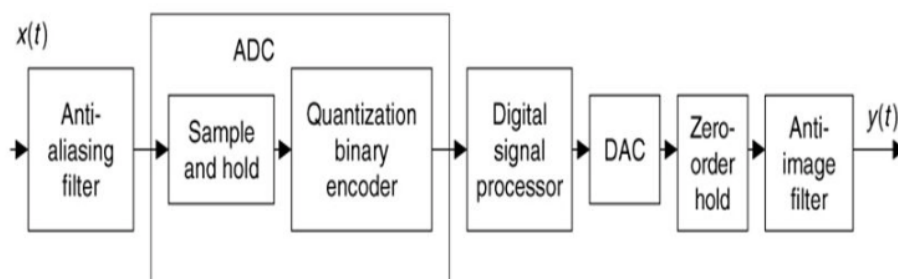
23. april 2017

**DTU**



**Danmarks  
Tekniske  
Universitet**

Denne kursusopgave er lavet i kurset 62516 - "Indledende medicinsk signalbehandling" på 2. semester af Sundhedsteknologi. Opgaven arbejder med anvendelsen af et DSP system ved brug i analysering af signaler. Der skal altså designes en række forskellige filtre, som senere skal bruges til "Digital Signal Processor" i DSP systemet. Dette skal denne opgave dog ikke omhandle. Disse hjemmelavede filtre (udført i Matlab) skal kunne fjerne støj fra og brum fra et signal. De skal dog også kunne detektere bestemte lyde i tale.



Figur 1: Et DSP system

Det grundlæggende koncept for et DSP-system er illustreret på ovenstående figur 1. Systemet består af forskellige dele, hvor et inputsignal,  $x(t)$ , betragtes. Den første box er det såkaldte ikke-ideele "anti-aliaserings lavpasfilter", hvor indgangssignalets lave frekvenser kommer igennem filtret, hvorimod de høje frekvenser, som overstiger Nyquist-kriteriet, dæmpes. Et ideelt "anti-aliaserings filter" fjerner alle de uønskede frekvenser og lader kun de relevante input frekvenser igennem. Det er dog ikke muligt, at have et ideelt "anti-aliaserings filter".

Den næste box i DSP systemet er en AD-konvertering (ADC). I denne sker først en "sample and hold", der låser de analoge værdier på et konstant niveau i en bestemt tidsperiode, hvilket giver os et trappe lignende signal. Efterfølgende skal disse værdier igennem "quantization binary encoder", som tager det analoge signal og kvantiserer dette til digitale værdier.

Næste box i DSP systemet er "digital signal processor"(DS), hvis opgave er at behandler det nu digitale signal.

Efter dette, skal det digitale signal konverteres tilbage til et analogt signal, hvor der anvendes en DA-konvertering (DAC). "Zero-order hold" udfører præcis samme funktion som "sample and hold", som også holder fast i den analoge værdi på et konstant niveau i en bestemt tidsperiode, hvilket giver os et trappe lignende signal.

Den sidste del af DSP-systemet er "anti-image filtret". Dette filter udglatte de trappetrins lignende værdier, således at kurven bliver gjort pæn og brugbar, så det ligner input-signalet mest muligt.

## Opgave 1: Lavpas-filter til støjreduktion (fjern noget hvid-støj fra et talesignal)

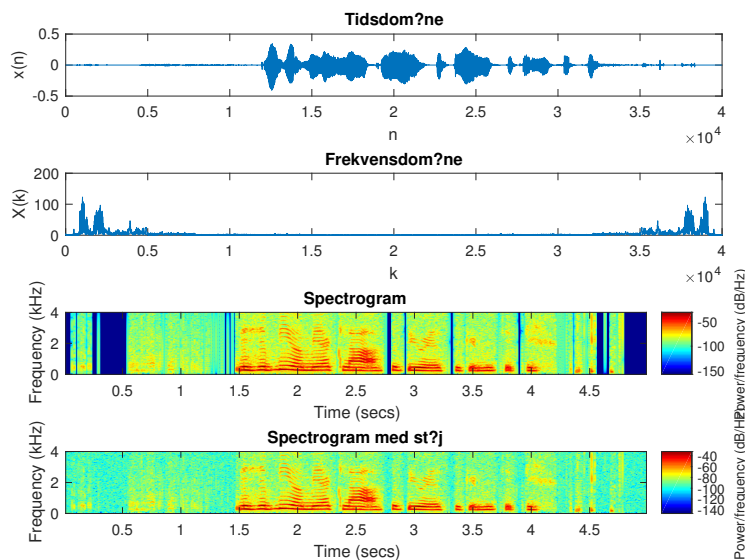
*Indspil selv et talesignal i Matlab og tilføj efterfølgende hvid støj til signalet i Matlab.*

*Analyser signalet før og efter tilføjelse af hvid støj – og bestem de nødvendige filterspecifikationer til et lavpas-filter, så kun de absolut nødvendige talesignaler (med støj oveni) kommer igennem filteret. Analyser signalet igen efter filtrering.*

Hvid støj defineres som et støjsignal hvor alle frekvenser repræsenteres med samme effekt. Ægte hvid støj er et uendeligt signal, der går fra tiden minus uendelig til plus uendelig med en uendelig samplingsfrekvens. I praksis er dette ikke muligt, og man arbejder derfor kun med et endeligt tidsdomæne hvor signalet har en begrænset båndbredde til den halve samplingsfrekvens (Nyquist).

Koden er rettet til, således et menneske kan sige den samme sætning på omkring 5 sekunder. - sætningen lød: "Hvis jeg bliver mere klar, så bliver jeg sgu gennemsigtig!".

Herunder ses talesignalet, hvor der er vist tidsdomæne, frekvensdomæne og henholdsvis to spektrogrammer, med og uden hvid støj.



Figur 2: Talesignal ved forskellige tilstande

Herunder vil der blive gennemgået lidt teori om SNR, da vi i Matlabkoden har anvendt funktionen `awgn(myRecording,SNR)` som tilføjer hvid Gaussisk støj til vektorsignalet, `myRecording`.

Signal-støj-ratio (SNR) er et værktøj, hvor niveauet af et ønsket signal sammenlignes med niveauet af hvid Gaussisk baggrundsstøj. SNR specificerer signal-støjforholdet pr. sample, målt i dB.

SNR er relateret til kvantiserings step. Dette forhold er givet ved:

$$E(e_q^2) = \frac{\Delta^2}{12} \quad (1)$$

Hvis man sætter  $x = myRecording$ , gives SNR ved formlen:

$$SNR = \frac{E(x^2)}{E(e_q^2)} = \frac{p_{signal}}{p_{stj}} \quad (2)$$

$$(3)$$

$$SNR = \frac{\sum_{n=0}^{N-1} x^2(n)}{\sum_{n=0}^{N-1} e_q^2(n)} \quad (4)$$

Via definitionen af decibel kan signal og støj udtrykkes i decibel (dB):

$$SNR_{dB} = 10 \cdot \log_{10}(SNR)dB = 10 \cdot \log_{10}\left(\frac{p_{signal}}{p_{stj}}\right)dB \quad (5)$$

Hvis man sætter  $E(x^2) = p_{stj} = x_{rms}^2$  fåes:

$$SNR_{dB} = 10.79 + 20\log_{10}\left(\frac{x_{rms}}{\Delta}\right)dB \quad (6)$$

man kan udregne  $x_{rms}$  og  $\Delta$  :

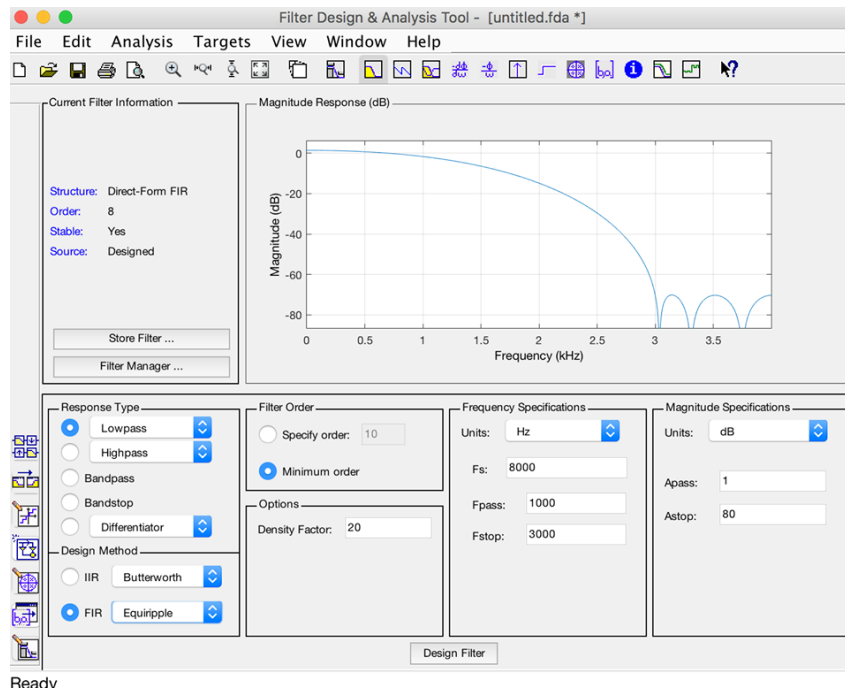
$$x_{rms} = \sqrt{\frac{1}{n} (x_1^2 + x_2^2 + \dots + x_n^2)}. \quad (7)$$

$$(8)$$

$$\Delta = \frac{x_{max} - x_{min}}{2^m} \quad (9)$$

Er forholdet større end 0 dB, angives der mere signal end støj. Jo højere SNR er, jo mere støj er det tilstede. Vi valgte at sætte  $SNR = 60dB$ , da man her kunne se signal og støj tydeligt. Vi prøvede værdier tættere på 0 dB, men fik for lidt støj. Vi prøvede ligeledes værdier højere en 60 dB, men her forsvandt signalet i støj.

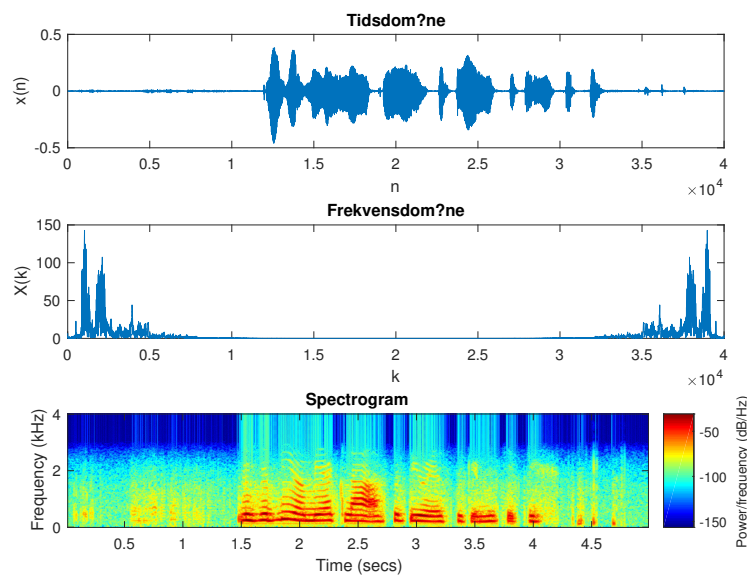
Herefter lavede vi et lavpasfilter vha. fdatool i Matlab. - vi valgte at generere et 8. ordens lavpas Fir filter, hvor samplingsfrekvensen er på 8000 Hz. Dette ses nedenfor:



Figur 3: Talesignal ved forskellige tilstande

For at køre talesignalet med hvid støj igennem vores lavpasfilter, gemte vi blot filterkoefficienterne fr fdatool i workspace.

Herunder ses talesignalet efter filteringen:



Figur 4: Talesignal ved forskellige tilstande

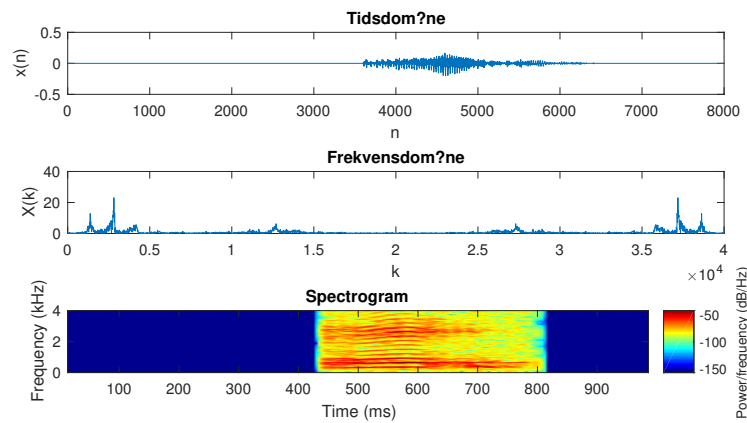
Det ses, at støjen er fjernet ved alt over cutof frekvensen på 3000 Hz. Hvorfor er der ikke cuttet af overalt?

## Opgave 2: Båndpas-filter til analyse af vokaler 'a' 'e' 'i'

Indspil selv lyd som indeholder vokalerne 'a' 'e' 'i' – analyser signalerne og bestem de nødvendige filterspecifikationer så f.eks. et 'a' detekteres i 'en talestrøm'. Implementer filteret i Matlab og dokumenter designet.

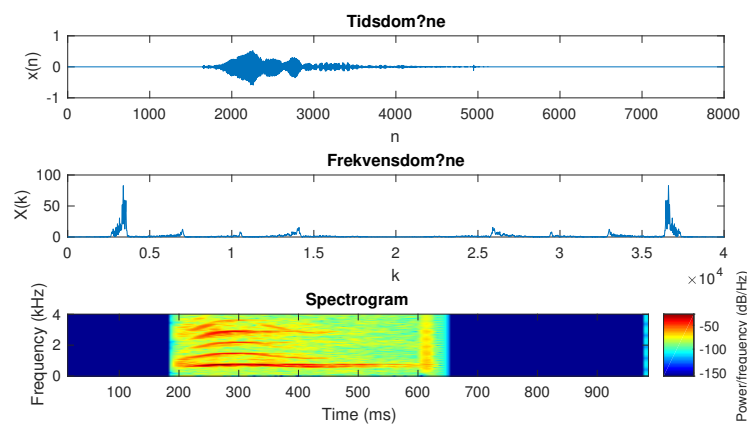
vi har valgt at indspille de 3 vokaler:

Et a:



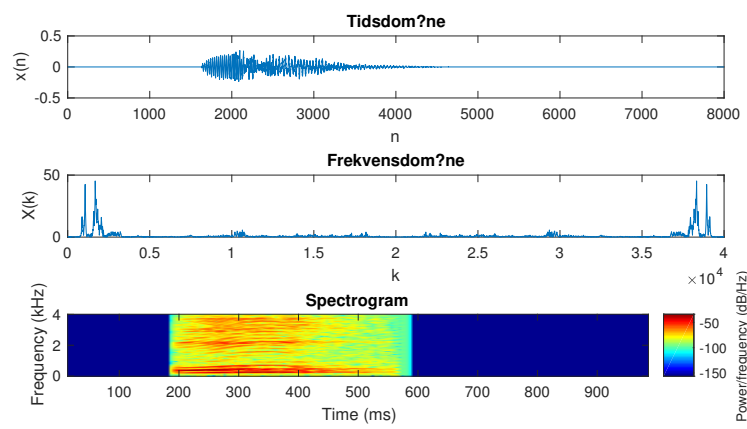
Figur 5: Talesignal ved forskellige tilstande

Et e:



Figur 6: Talesignal ved forskellige tilstande

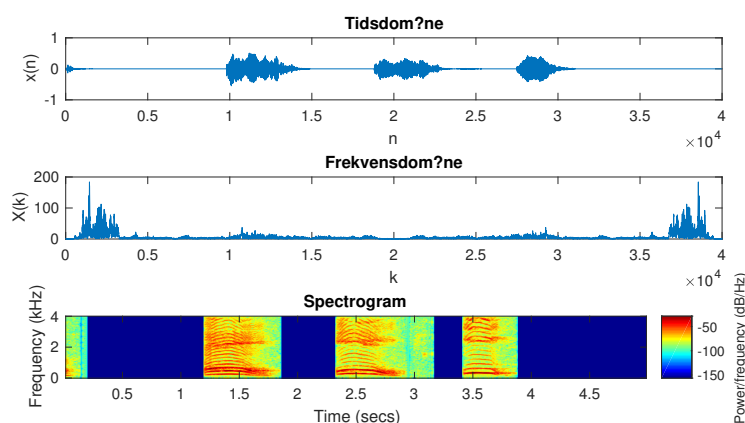
Et i:



Figur 7: Talesignal ved forskellige tilstande

alle 3 signaler i en:





Figur 8: Talesignal ved forskellige tilstande

### Opgave 3: Båndstop filter til fjernelse af brumsignal

I folderen '(Signaler til opgave3') ligger et signal med indlejret brum. Analyser signalet og bestem de nødvendige filterspecifikationer så brummen fjernes. Implementer filteret i Matlab og dokumenter designet

### Opgave 4: Filter til at finde triggerpunkt for EKG signal (ekg signal fra øvelse 1)

Anvendt EKG signalet fra øvelse 1 – og proces signalet gennem en simpel differentiering, så man kan finde et passende 'triggerpunkt' til at bestemme bestemte steder på signalet. Implementer filteret i Matlab og dokumenter designet.

### Opgave 4: Filter til at finde triggerpunkt for EKG signal (ekg signal fra øvelse 1)

Anvendt EKG signalet fra øvelse 1 – og proces signalet gennem en simpel differentiering, så man kan finde et passende 'triggerpunkt' til at bestemme bestemte steder på signalet. Implementer filteret i Matlab og dokumenter designet.

**Opgave 5:** Anvend en vindue-funktion på et af ovenstående signaler. Forklar hvorfor en vindue-funktion kan være nødvendig, hvad den gør og hvordan den anvendes.

*Vis ved et eksempel*