

INTRODUKTION TIL DIGITAL SIGNALANALYSE

E4DSA

Janus Bo Andersen (JA67494)

4. juni 2020

EMNER

15 MIN. + VOTERING

Spørgsmål

1. Frekvensanalyse med DFT + case 1
2. Stokastiske signaler og SNR
3. FIR-filterdesign m. /vinduesmetoden
4. IIR-filterdesign m. pol-/nulpunktsplacering + bilineær z-trf. + case 2
5. Stokastiske signaler og midlingsfiltre + case 3
6. Decimering og interpolation.
7. Foldning og korrelation + case 4
8. Differentiation og integration

1. FREKVENSANALYSE MED DFT

1. Overblik over DFT'en
2. Eksempel
3. Samplingsteorement (Shannon/Nyquist)
4. Eksempel 2:
 1. Spektral lækage
 2. Vinduesfunktioner, Zero-padding
 3. Mere data
5. Spektrogrammet, STFT'er
6. Overblik over vinduesfunktioner (hvis der er tid)
7. Opsummering.

Spørgsmål 1

Diskutér frekvensanalyse med DFT. Herunder frekvensopløsning, zero-padding, vinduesfunktioner og spektrogram. Referér til Case-projekt 1.

DFT



Formål

Den Diskrete Fouriertransformation (DFT)

- Transformation $x(n) \rightarrow X(m)$ fra tidsdomæne til frekvensdomæne
- Invers transformation $X(m) \rightarrow x(n)$ rekonstruerer signal i tidsdomæne
- Benyttes på samplet signal, data, $x(n)$ med længde N så $n = 0, \dots, N - 1$
- "Altid" via FFT-algoritmen.
 - DFT er $\sim O(N^2)$, FFT er $\sim O(N \log_2 N)$. Med $N = 1000$, fås faktor x100.

DFT viser frekvensindhold i et samplet signal

- Antag samplet med samplingsfrekvensen f_s [Hz]
- $X(m)$ repræsenterer mængde af frekvensindhold ved frekvensbin m :
- Frekvensakser $\omega(m) = \frac{2\pi}{N}m$ [rad/sample] (normaliseret) $\leftrightarrow f(m) = m \frac{f_s}{N}$ [Hz]
- Frekvensopløsning:** $\Delta f = \frac{f_s}{N}$
- $X(m)$ er kompleks koefficient for Fourierrække, dvs. både *cos* og *sin*.

DFT kan også ses som

- Basisskitte, ortogonal projktion i N -dimensionelt vektorrum (indre produkter)
- Diskret sampling af DTFT
- Funktionsapproksimation (med sin/cos i stedet for poly i fx Taylorrækker)

Formler

$$X(m) = \sum_{n=0}^{N-1} w(n)x(n)e^{-j\frac{2\pi m}{N}n}$$

Annotations for DFT formula:

- DFT, analyse: m 'te frekvenssample
- Summation over N samples
- Indre produkt med den komplekse eksponentialfunktion ($e^{j\omega} = \cos\omega + j\sin\omega$)
- Normaliseret frekvens fra 0 til $\frac{2\pi(N-1)}{N}$ [rad/sample]
- Evt. vinduesfunktion
- n 'te sample fra samplet signal
- n 'te "sample" fra eksponentialfkt.

Annotations for IDFT formula:

- IDFT, syntese: n 'te signalsample

$$x(n) = \frac{1}{N} \sum_{m=0}^{N-1} X(m)e^{j\frac{2\pi n}{N}m}$$

EKSEMPEL 1:

Værd at bemærke – vi kan dekomponere et signal

Samplede signaler:

- $N = 512$ samples med $f_s = 1024$ Hz
- Samples ved $n = 0, \dots, N - 1$, dvs. $t = nT_s = \frac{n}{f_s}$

Benytter FFT(!) til at beregne DFT:

- Frekvensopløsning: $\Delta f = \frac{f_s}{N}$,
- Analysefrekvenser: $f(m) = m \Delta f$
- f_0 og f_1 ligger lige på bins: Ingen lækage, ingen scalloping loss

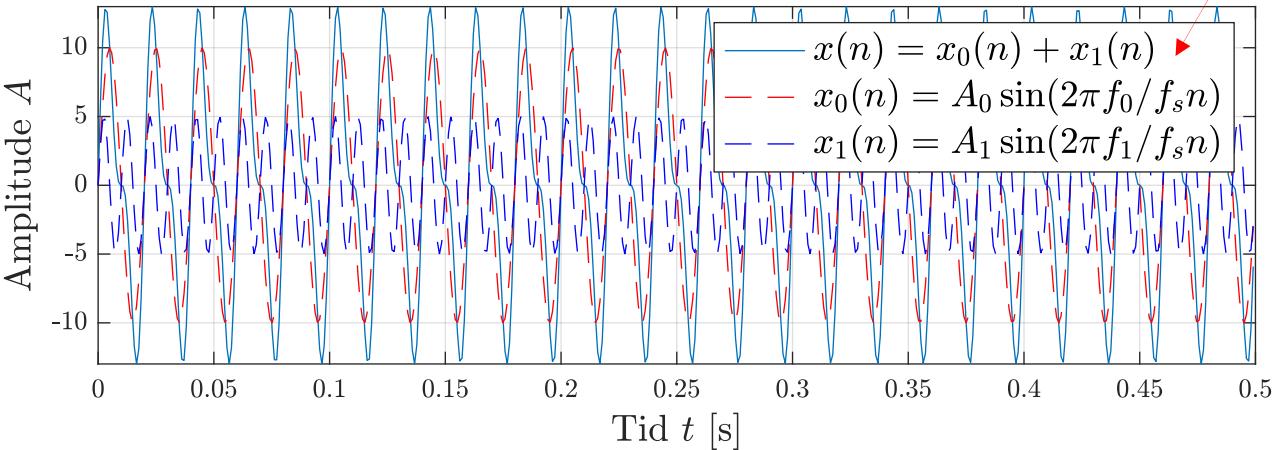
Skalering:

- $\frac{2}{N} |X(m)|$ giver repræsentation af amplituder fra tidsdomæne
 - Faktor 2 pga. spejling i Nyquist, $\frac{1}{N}$ pga. IDFT

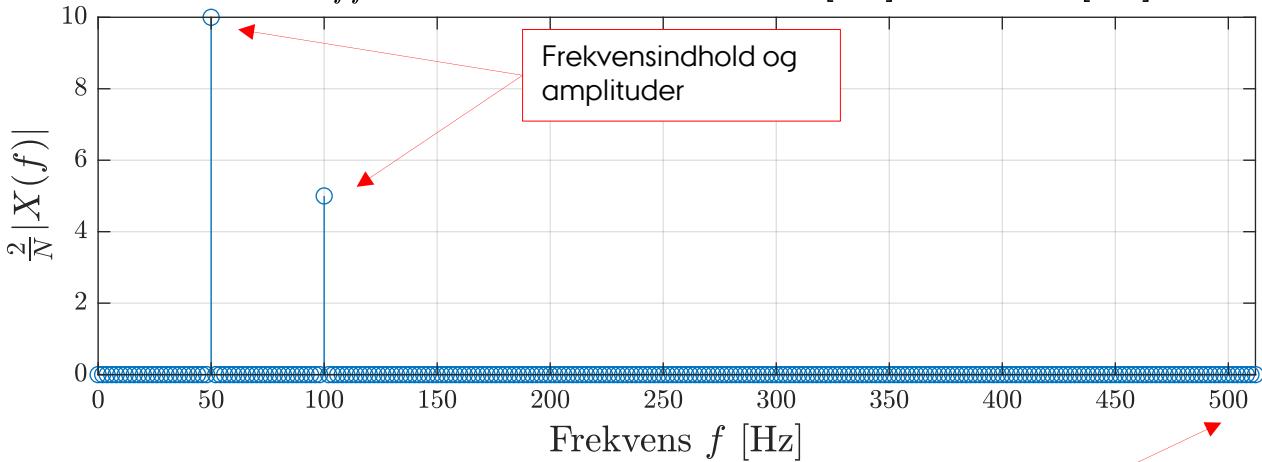
Shannon/Nyquist samplingsteorem:

- $f_s \geq 2B$, og sig. kan repræsenteres fuldt, hvor B er højeste frekvens i sig.
- Dvs. giver mening at se på op til $\frac{f_s}{2}$ [Hz]
- Højere frekvenser aliaseres.

Tidsdom.: $A_0 = 10$, $f_0 = 50$ [Hz], $A_1 = 5$, $f_1 = 100$ [Hz]
 $N = 512$, $f_s = 1024$ [Hz],



FFT: $N_{fft} = N = 512$, $f_s = 1024$ [Hz], $\Delta f = 2$ [Hz]



SAMPLINGSTEOREM

Værd at bemærke

Shannon/Nyquist samplingsteorem:

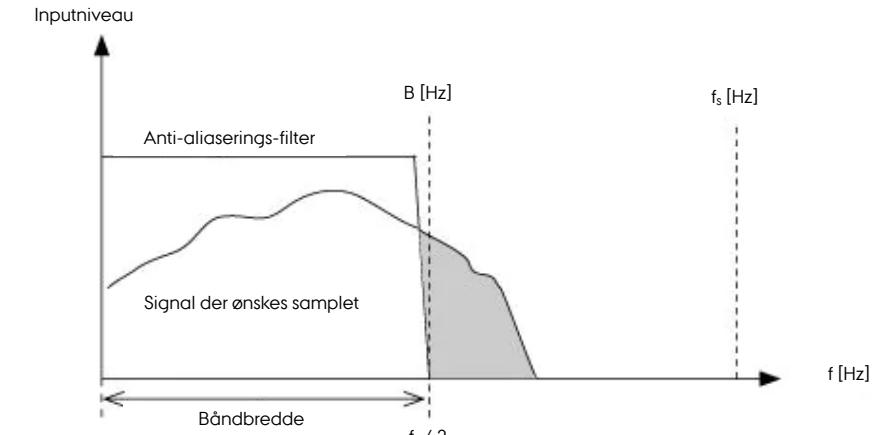
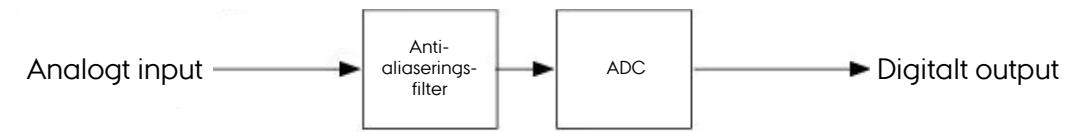
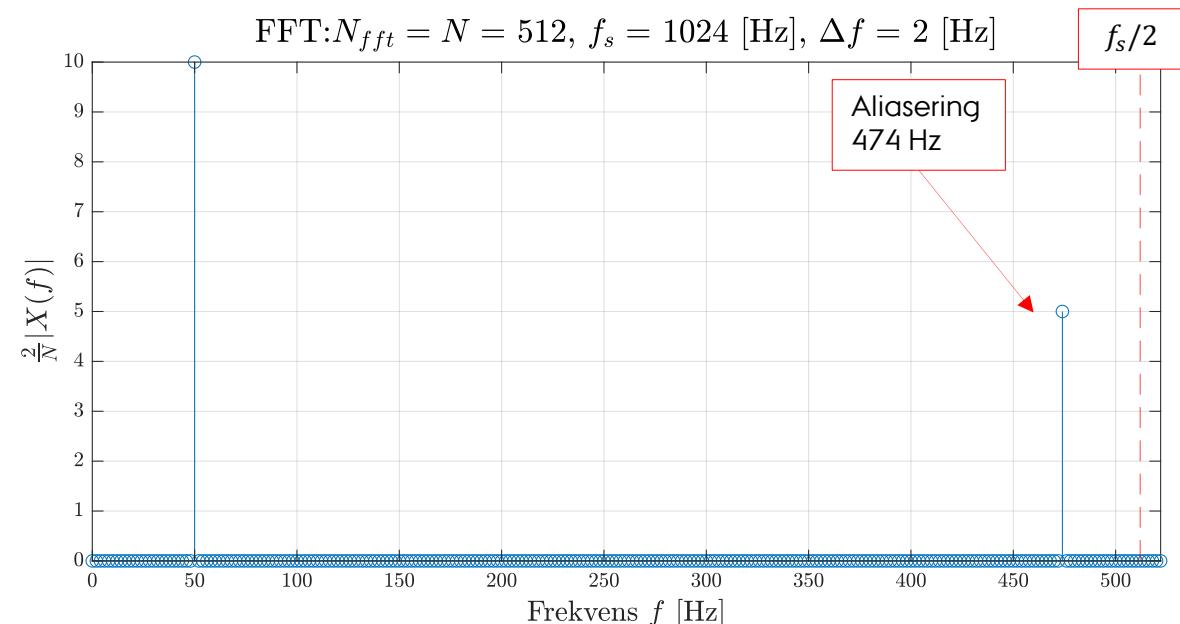
- $f_s \geq 2B$, og sig. kan repræsenteres fuldt, hvor B er højeste frekvens i sig.
- Dvs. giver mening at se på op til $\frac{f_s}{2}$ [Hz]
- Højere frekvenser aliaseres.
- Se figur 1: En frekvens højere end $f_s/2$ aliaseres ind
- Altid anti-alisering før ADC!
- Ellers rekonstruerer vi et helt forkert signal!

Eksempel:

$$f_s = 512 \text{ Hz}$$

$$x(n) = 10 \sin(2\pi 50nT_s) + 10 \sin(2\pi 550nT_s)$$

Bliver spejlet i Nyquist-frekvensen:
550 Hz - 512 Hz = 38 Hz
512 Hz - 38 Hz = **474 Hz**



EKSEMPEL 2:

Værd at bemærke

Flytter f_0 et halvt bin -> ligger mellem frekvensbins

- $f_1 = 2f_0$ ligger stadig lige i et bin.

Spektral lækage

- Energien i signalet "lækker" ind i tilstødende bins
- Spektret bredes ud
- "Nye" frekvenskomponenter fremtræder

Scalloping loss

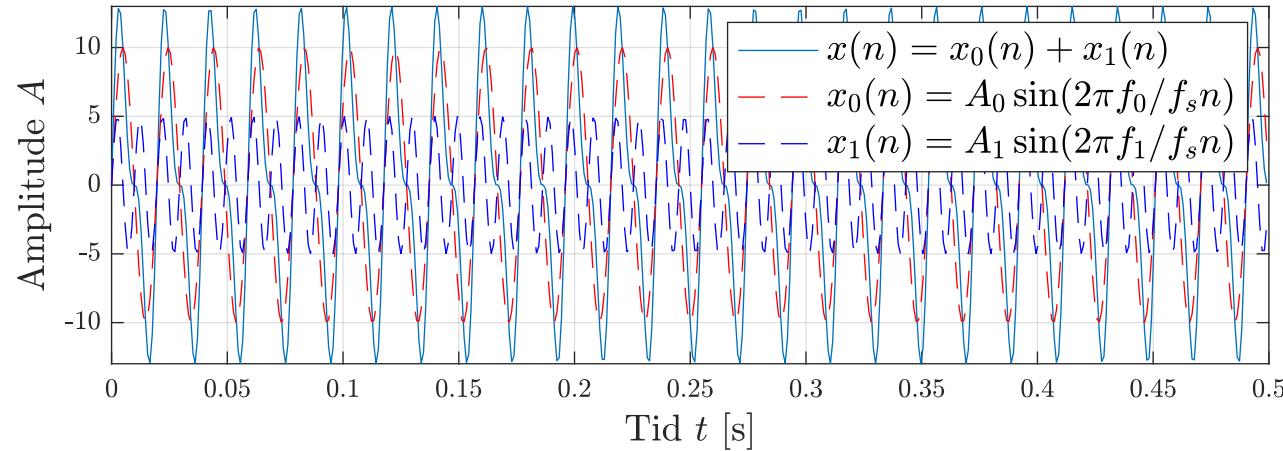
- Frekvens mellem to bins
- Frekvenskomponentens effekt deles ud over flere bins

Hvad kan man evt. gøre?:

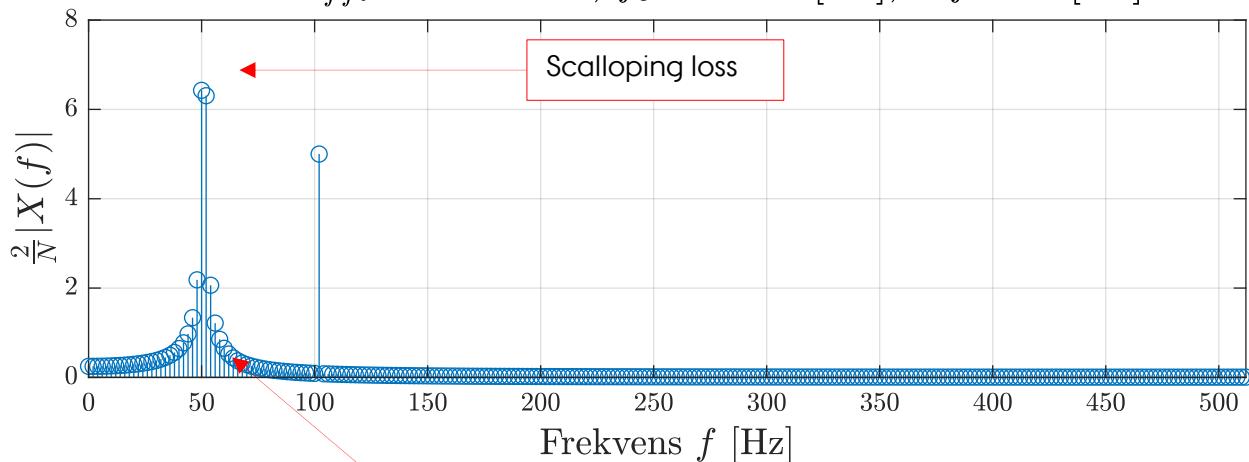
- Vinduesfunktioner
- Zero-padding -> øge den *tilsyneladende* frekvensopløsning
- Flere samples, mere data -> øge den faktiske frekvensopløsning.

Ej på frekvensbin

Tidsdom.: $A_0 = 10, f_0 = 51 \text{ [Hz]}, A_1 = 5, f_1 = 102 \text{ [Hz]}$
 $N = 512, f_s = 1024 \text{ [Hz]},$



FFT: $N_{fft} = N = 512, f_s = 1024 \text{ [Hz]}, \Delta f = 2 \text{ [Hz]}$



Spektral lækage

EKSEMPEL 2:

Værd at bemærke

Samme signal som før

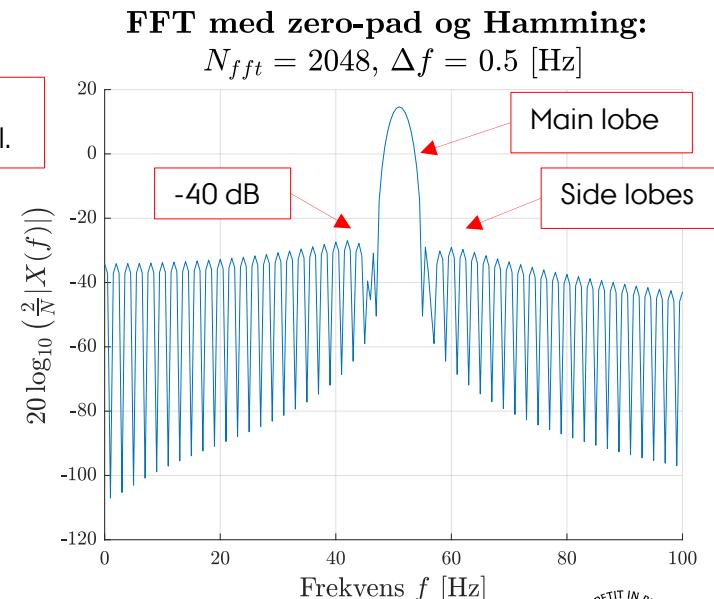
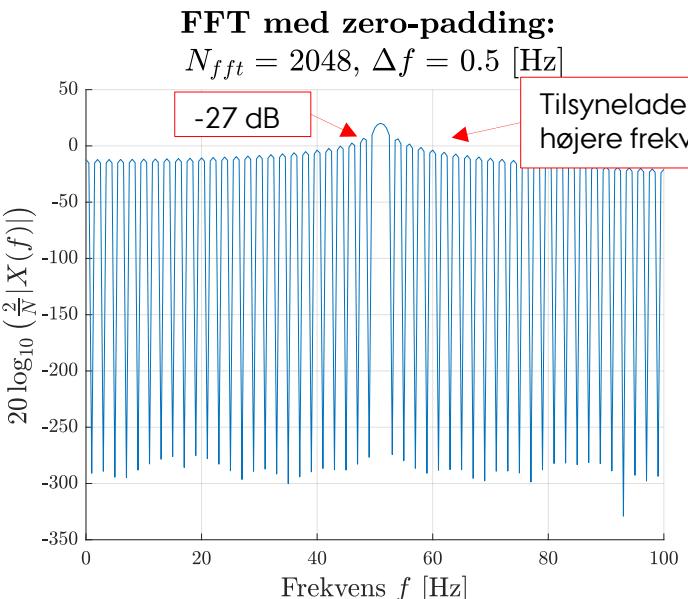
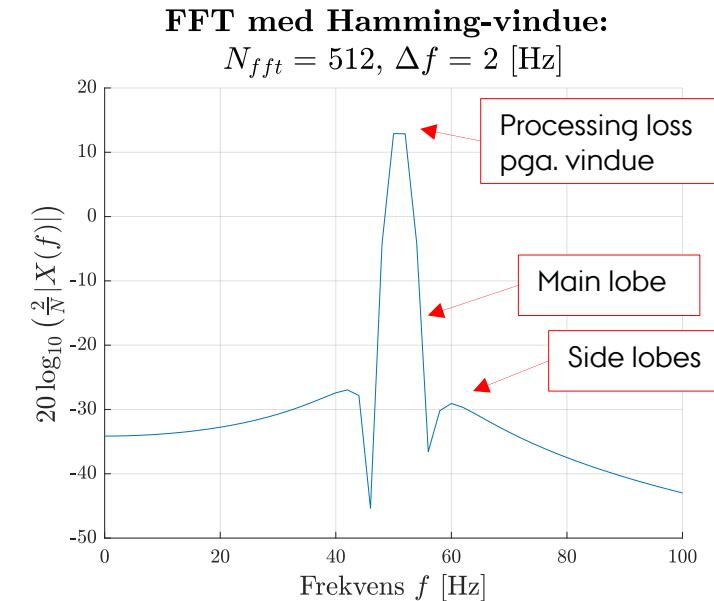
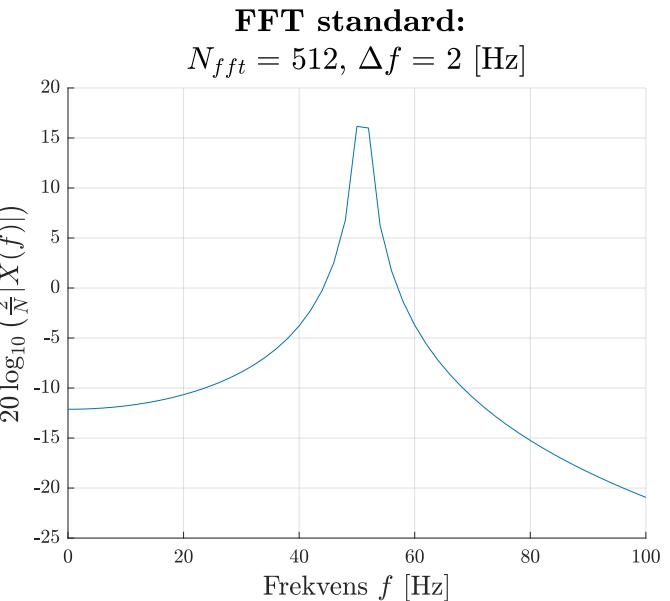
- $N = 512, f_s = 1024 \text{ Hz}$
- Nu vises effektspektrum: $20 \cdot \log_{10} \left(\frac{2}{N} |X(f)| \right)$

Zero-padding:

- Tilføjer nuller i enden af signalet (1536)
- Øger tilsyneladende frekvensopløsning
 - Men tilføjer ikke ny information -> værdier ift. gamle bins er de samme
 - Blot flere samlinger af DTFT
- Det er en interpolation. Nem måde at interpolere.

Vinduesfunktion – her Hamming-vindue:

- Bredere main lobe
- Yderligere processing loss pga. vindue
- Dæmper sideslobes <- som ønsket!



BEDSTE LØSNING: MERE DATA!

Værd at bemærke

Som før

- $f_0 = 51$ Hz.
- 4 gange så mange samples i stedet for x4 zero padding

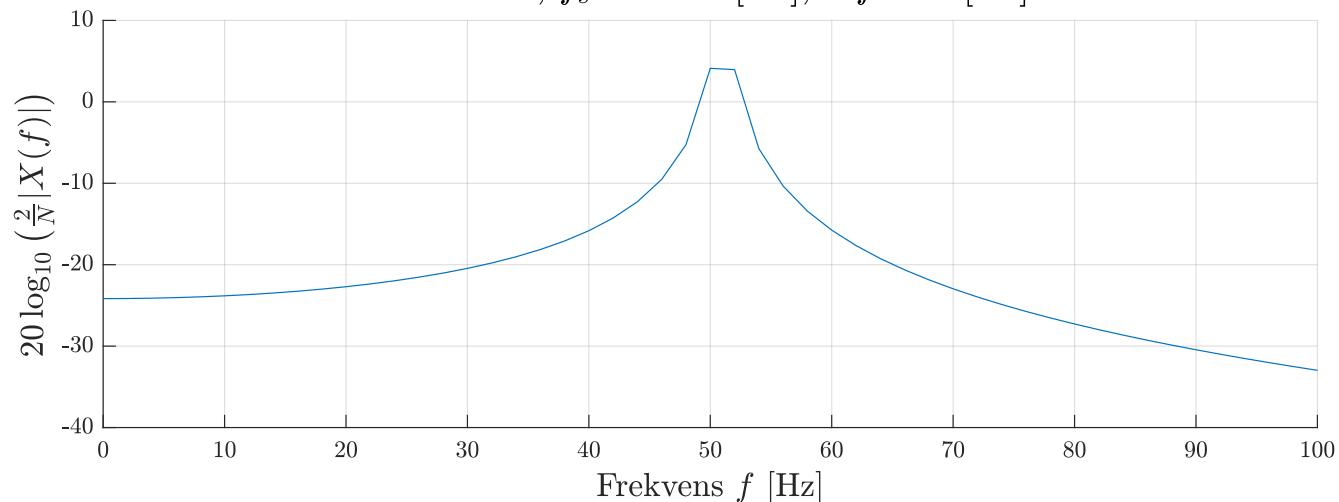
Allerbedste løsning: Indhent mere data

- Bedre reel frekvensopløsning
- Smallere main lobe
- Mere dæmpede

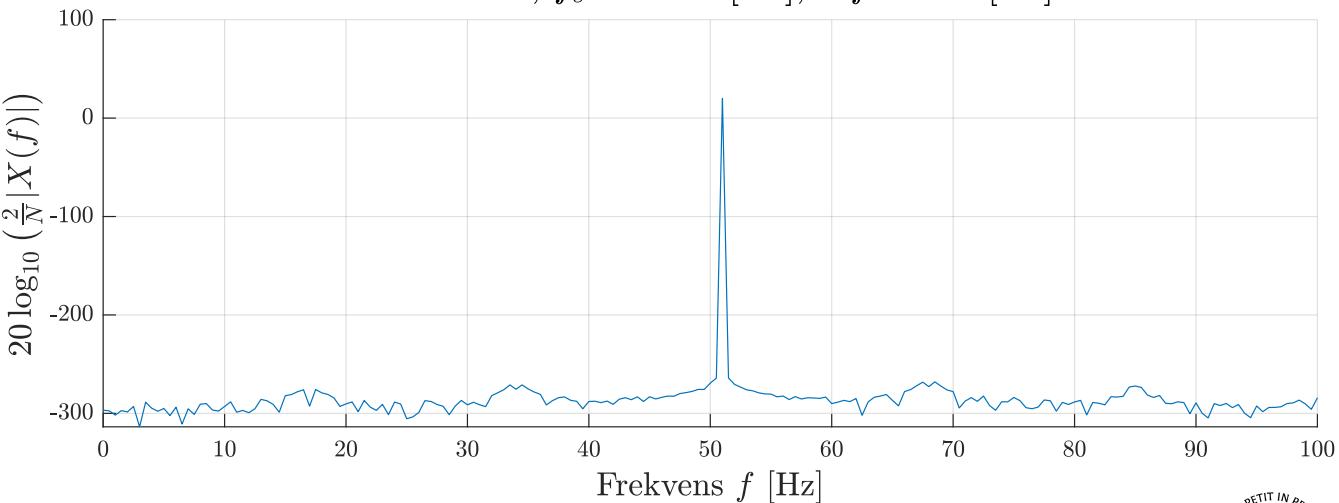
Kræver naturligvis et stationært signal

- Apropo -> spektrogram...

FFT som tidligere:
 $N = 512, f_s = 1024$ [Hz], $\Delta f = 2$ [Hz]



FFT med 4 gange flere samples:
 $N = 2048, f_s = 1024$ [Hz], $\Delta f = 0.5$ [Hz]



VINDUESFUNKTIONER

Værd at bemærke

Valg af vinduer er et trade-off:

- Frekvensspektrum: Bredere main lobe vs. højere dæmpning af side lobes
- Filtrering: Bedre dæmpning i stopbånd vs. bredere transitionsbånd (evt. $N \uparrow$)
- Udvalgte typer:
 - Rektangulær (intet): -13 dB
 - Hann: -31 dB
 - Hamming: -41 dB
 - Blackman -57 dB

Hvorfor virker et vindue?

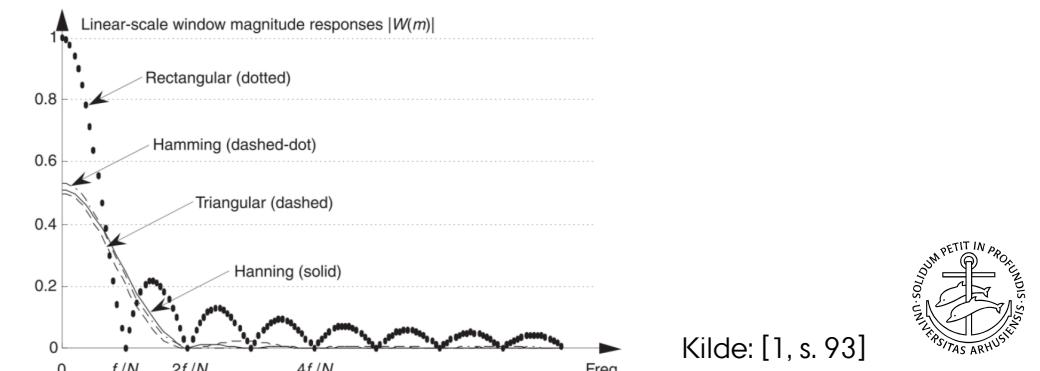
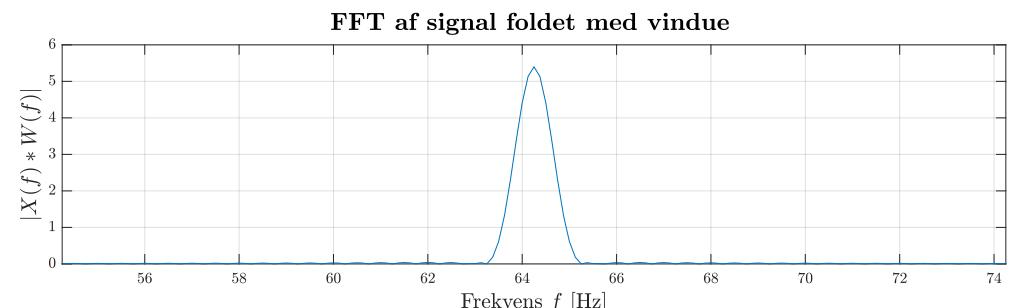
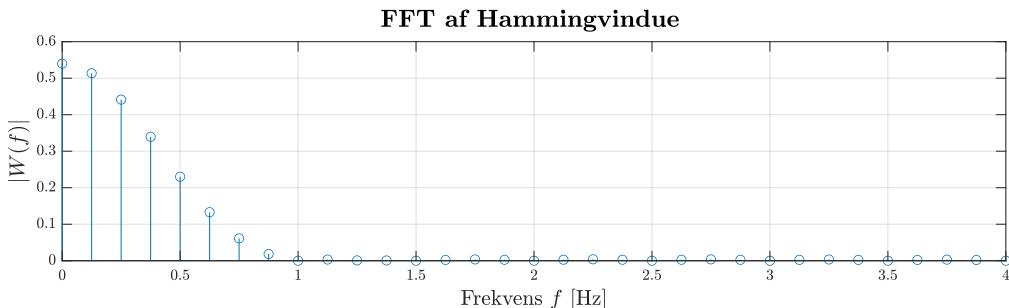
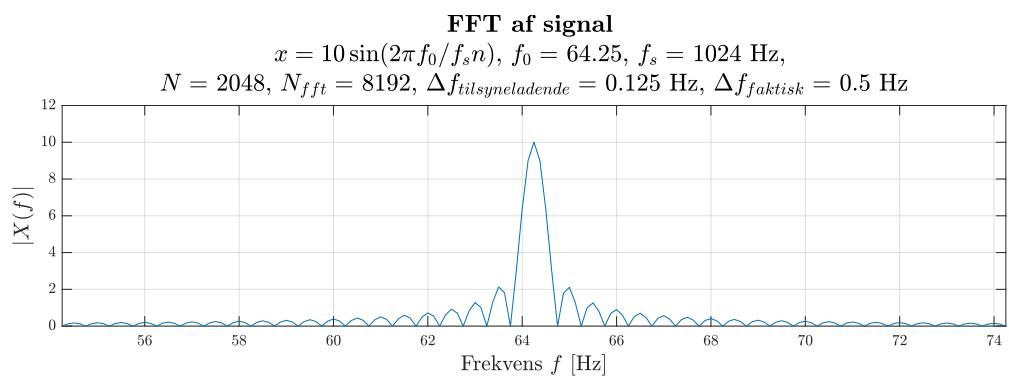
I tidsdomænet:

- Diskontinuiteter ved ende af samplinginterval, fordi der ikke er samplet et heltal perioder. Diskontinuitet dæmpes af vindue
 - d.s.s. at en given frekvens ligger mellem to bins

I frekvensdomænet:

- Vindue virker som et "smalt" filter: Bytter "bredde" for "amplitude"
- Selekterer kun nærmeste frekvensbins i foldningssummen, og dæmper dem.
- Kan også ses som amplitudemodulation.

Ulempe: "Processing loss" (reduceret peak amplitude)



STDFT - SPEKTROGRAM

Formål og anvendelse

Analyse af ikke-stationære signaler

- Frekvensindhold varierer som en funktion af tid
- Rulle DFT'en hen over et tidssignal

Short-time Discrete Fourier Transform (STDFT):

- Bryder signalet op i segmenter, og laver DFT på hvert segment
- Vælg længde af segment (vindueslængde), L
- Vælg stepstørrelse, Δn
- Tilføj evt. et vindue, fx Hamming, $w(n)$

Trade-off i længde af vinduet, L :

- Et langt vindue giver finere frekvensopløsning: $\Delta f = \frac{f_s}{L}$
- MEN, grovere tidsopløsning: $T_{STDFT} = L T_s = \frac{L}{f_s}$.
- De to mål er altå inverse, med trade-off i at have finmasket adskilning i *enten* tid *eller* frekvens.

Formler

DFT, analyse: k 'te frekvenssample

Vindueslængde, L

$$X(k, n) = \sum_{m=0}^{L-1} w(m)x(n + m)e^{-j\frac{2\pi km}{N}}$$

Summation over N samples

Segment omfatter samples:
 $n, \dots, n + L - 1$

EKSEMPEL FRA CASE 1

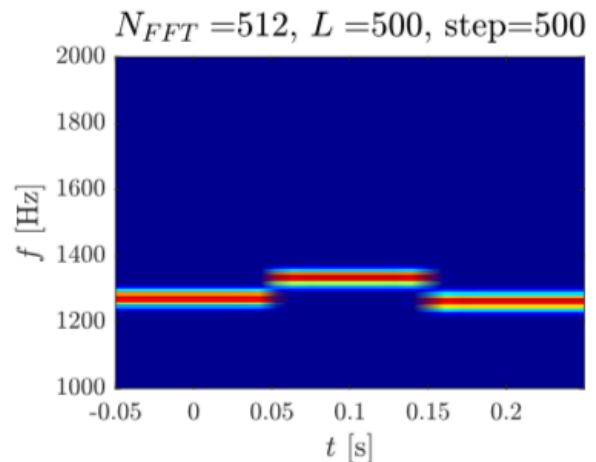
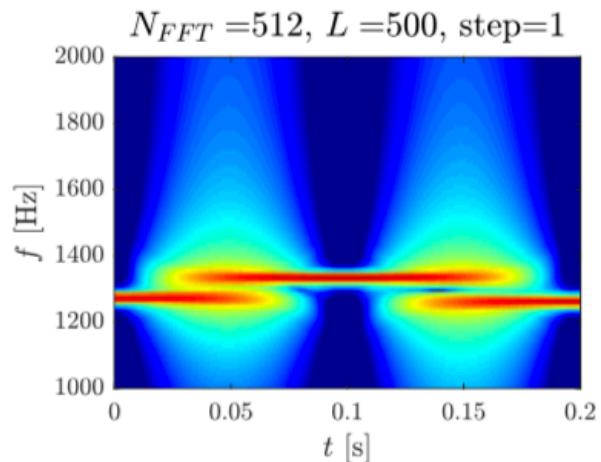
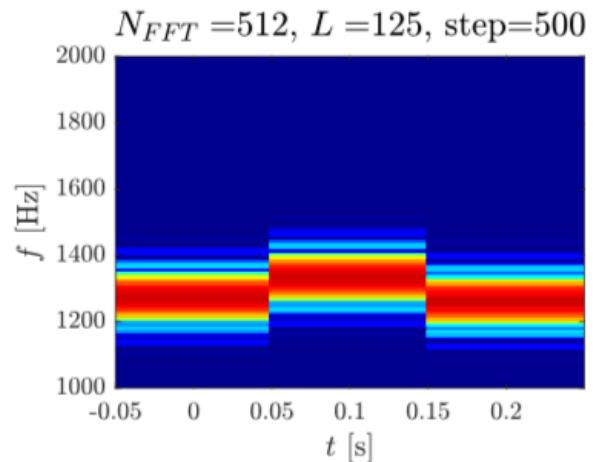
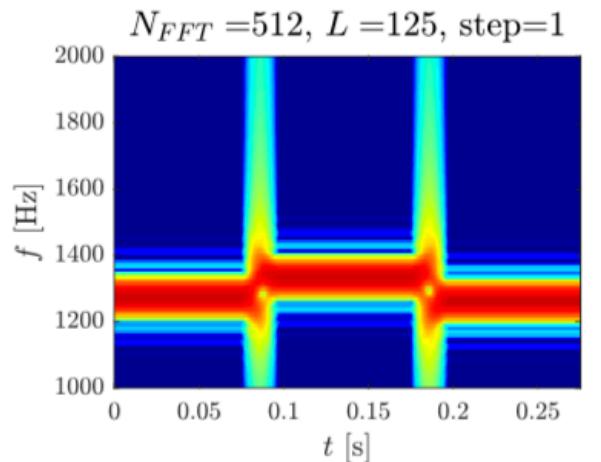
Formål og anvendelse

Eksempel (case 1):

Sender signalet 'DSA' med FSK (frekvens sat ift. ASCII)

- Baudrate 10 symboler/sek. -> ikke-stationært signal, skal afkodes
- Spektrogrammer illustrerer trade-off:
 - Øverste række t.v.:
 - kort vindue, klar separation i tid for hvornår en frekvens er indeholdt i signalet eller ej -> fin tidsopløsning.
 - Men, frekvensspektret bredt, altså grov frekvensopløsning. Svært at pin-pointe en eksakt frekvens, eller at adskille flere tætliggende frekvenser.
 - I nederste række t.v.:
 - Med langt vindue, er situation omvendt.
- Spektrogrammer t.h. illustrerer, at forhåndsinformation om signalet er værdifuldt.
 - Stepstørrelsen er tilpasset antallet af samples for hvert symbol.
 - Tydelig separation i tid, og med et langt vindue fås også en fin frekvensopløsning.

Spektrogram



OPSUMMERING

Gennemgået lidt teori omkring frekvensanalyse med DFT og vinduesfunktioner
DFT'en – i form af FFT - ekstremt anvendeligt indenfor videnskaberne

- Dekomponere signaler / signalanalyse
- Adskille signal og støj
- Data-/telekommunikation
- Billedkompression

Øvrig perspektivering:

- Effekt og energi i et signal – *Parsevals sætning*.
- Støj og stokastiske signaler, PSD
- FIR-filterdesign med vinduer

2. STOKASTISKE SIGNALER OG SNR

1. Stokastiske signaler
2. Parsevals sætning
3. Signal vs. støj (SNR)
4. Case 1: FSK-transmission
 1. Designvalg
 2. SNR-eksperiment
5. Lidt mere om stokastiske signaler
(hvis der tid)
6. Opsummering

Spørgsmål 2

Diskutér stokastiske signaler og SNR. Herunder måling af SNR i tids- og frekvensområdet.

STOKASTISKE SIGNALER

Signaler med stokastiske komponenter. Stokastisk: Som ikke kan forudsiges præcis, muligvis med en sandsynlighedsfordeling. Def. af støj: Uønsket fluktuation i data. Fx additiv støj, hvor observerbar data = signal + støj:

$$x(n) = \mu(n) + \epsilon(n)$$

Støjen kan fx være givet ved $\epsilon \sim N(0, \sigma^2)$ (hvid støj).

Motivation:

- Sensordata med støj (støjgulv)
- Målinger med måleusikkerhed
- Osv.

Type

Tidsdomæne

Stationære
signaler

Tids-asyriskron midling i tidsdomæne reducerer støj på estimat af "sand" middelværdi

- MA-filtre, eksponentielle midlingsfiltre

Ikke-
stationære
signaler

Tids-synkron sampling (med faseinformation)

- Parallel-midling

Opdele signal i stationære segmenter

Spektralanalyse

Power Spectral Density med reduceret støj:

- Ikke-parametrisk med fx Welch's metode
 - Midling af adskillige FFT'er dæmper overlejret støj
- Parametrisk med fx en LS-fitted AR-model
 - PSD regnes pba. a -koefficienterne.

Model så proces gøres stationær (fx first-difference)

Spektrogram

Stationære sektioner, midling i frekvensdomæne

BEREGNING AF EFFEKT I ET SIGNAL

Parsevals sætning

- Total energi i et signal kan findes ved at summere

- Effekt-per-sample over *tid*

- Spektral effekt over *frekvenser*

Tidsdomæne

Frekvensdomæne

$$\sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2$$

SIGNAL VERSUS STØJ (SNR)

"Udfordring" med stokastiske signaler:

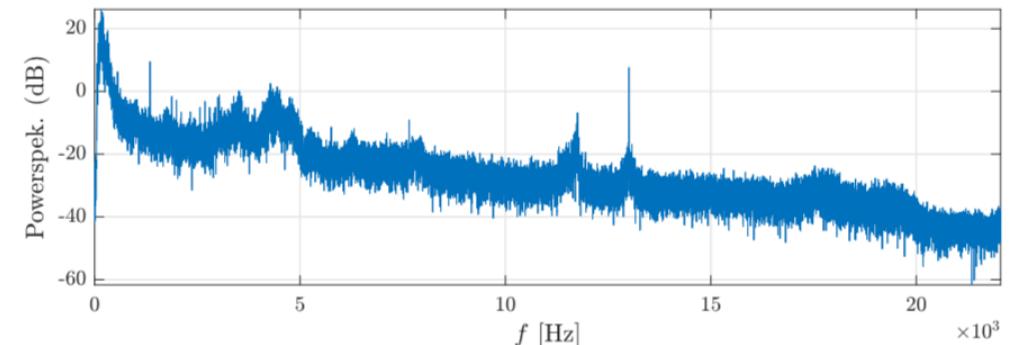
- Adskille "signal" fra "støj"
- Ikke-parametrisk metode: **Effektspektrum (PSD)**
- Parametriske metoder: Lineære modeller, AR, osv.

$$\text{SNR} = \frac{\text{Signal power}}{\text{Noise power}} = \frac{\text{Sum af } |X(k)|^2 \text{ for signal}}{\text{Sum af } |X(k)|^2 \text{ for støj}}$$

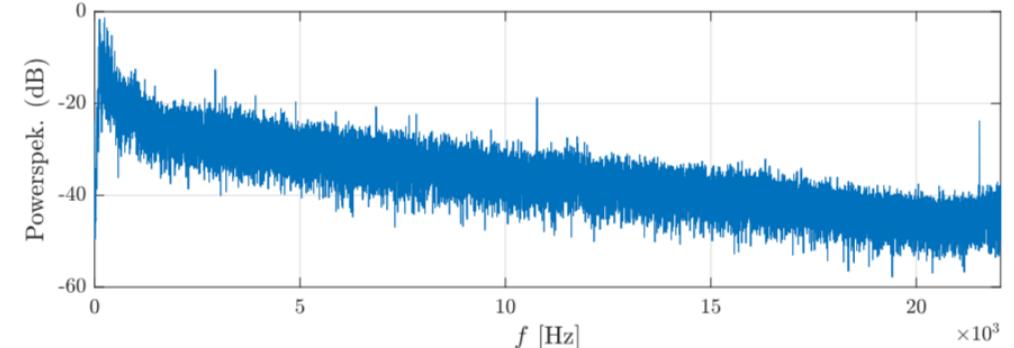
$$\text{SNR}_{dB} = 10 \log_{10}(\text{SNR})$$

Husk: Der er altid et støjgulv (se til højre).

Støjgulv og sammenligning af mikrofoner
Frekvensindhold i omgivelser med indbyg. mic.



Frekvensindhold i omgivelser med ekstern mic.



CASE 1: FSK-TRANSMISSION

Kort fortalt...

Kommunikation vha. Audio FSK (Frequency Shift Keying)

- Valg af transmissionsbånd (versus medium og udstyr)
- Enkodering af symbolsæt til frekvenser i båndet (ASCII – men måske ikke bedste idé)
- Valg af Baudrate

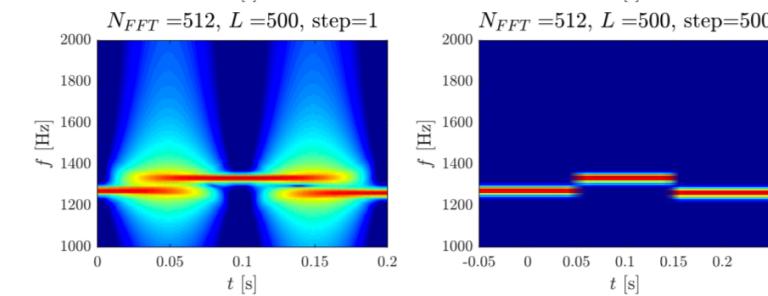
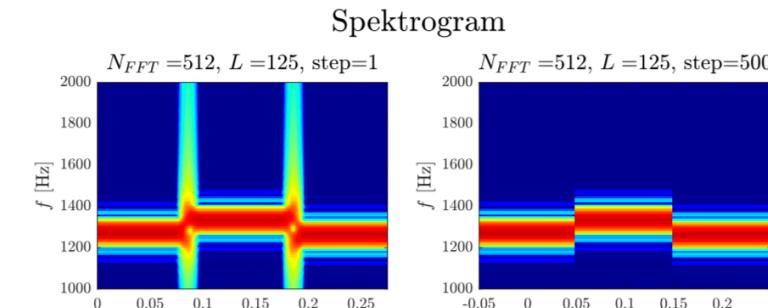
$$T_{sym} = \frac{1}{baudrate}$$

Udfordringerne er

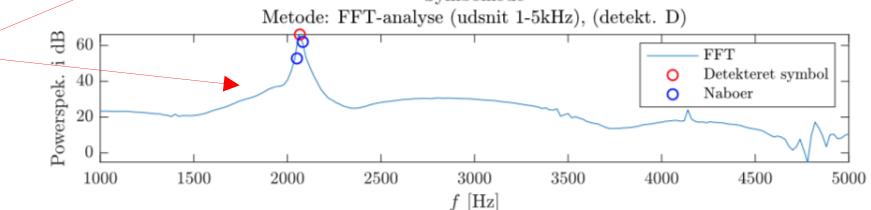
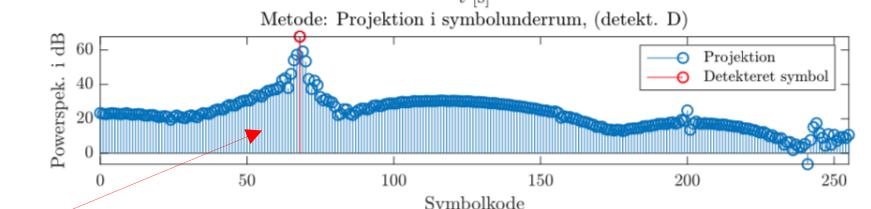
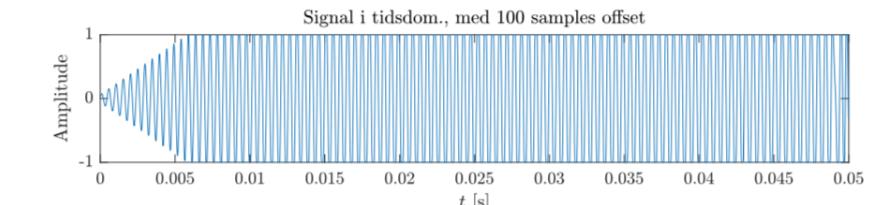
- Hvor langt og hvor hurtigt kan man transmittere?
- Hvor meget støj kan dekodning håndtere (DFT el. lign.)
- **Funktion af SNR**

Udvælge "signal" via bins eller tærskelværdi

Estimation af SNR (dB):
Projektionsmetode: 4.5946 dB
FFT-metode: 2.6492 dB



Transmission med 20 baud



EKSPERIMENT

Kort fortalt...

Setup:

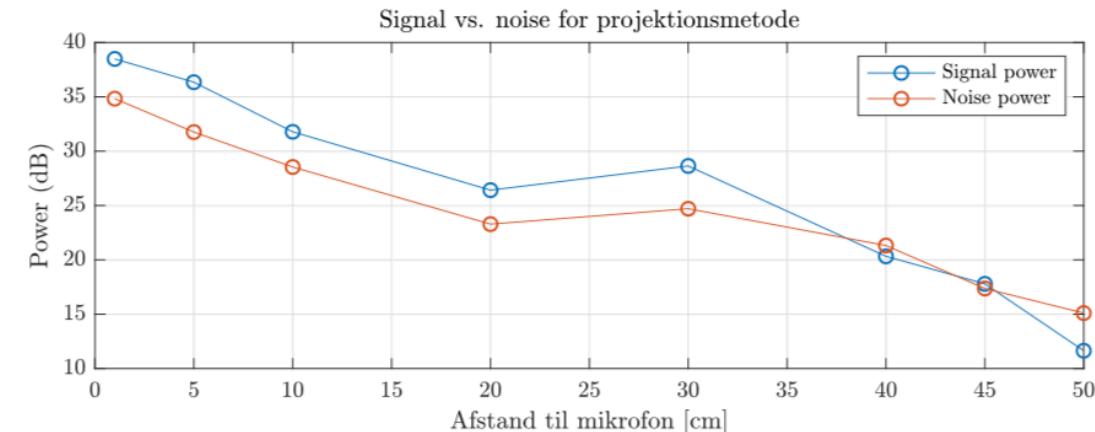
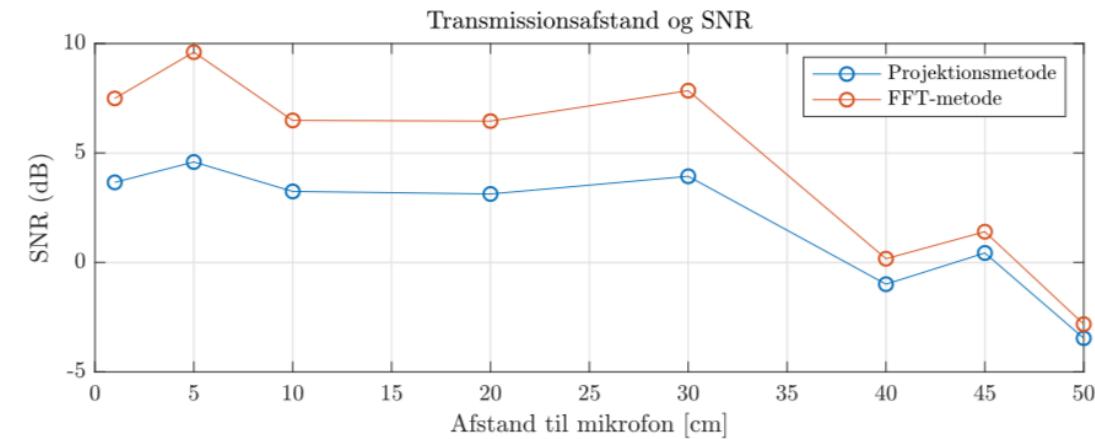
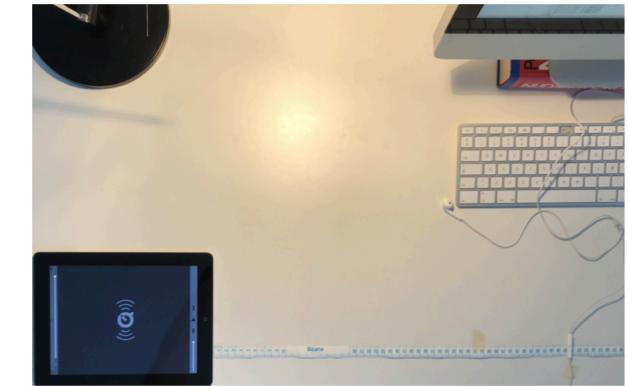
- 1-5 kHz bånd, 20 baud, besked på 44 symboler.

Øger afstanden mellem højttaler og mikrofon

- Over hvor lang afstand kan der kommunikeres korrekt?
- Ser udvikling i SNR

Resultater

- Ved 50 cm opstår fejl
 - $\text{SNR}_{\text{dB}} < 0$
 - Effekt fra støj (støjguld) væsentligt større end fra signalet (4 dB over)
- Lydtryk falder af med $1/r$, støjguld konstant.



HVAD KUNNE MAN ELLERS GØRE?

Midling i frekvensdomænet (PSD)

Tids-synkron midling i tidsdomænet (med faseinformation)

- Fx parallelle modtagere

Bedre udstyr, andet frekvensbånd, osv.

ANDRE VINKLER PÅ STØJ

Type

Formel

Middelværdi

$$\bar{x} = \frac{\sum_{n=0}^{N-1} x(n)}{N}$$

Varians på
middelværdi

$$Var(\bar{x}) = \sigma_{\text{output}}^2 = \frac{\sigma_{\text{input}}^2}{N}$$

Varians

$$\hat{\sigma}^2 = \frac{\sum_{n=0}^{N-1} (x(n) - \bar{x})^2}{N}$$

Standard-
afvigelse
(spredning)

$$\hat{\sigma} = \sqrt{\hat{\sigma}^2}$$

Betydning

DC-niveau eller DC-offset
Estimat på den sande middelværdi i en
dataserie

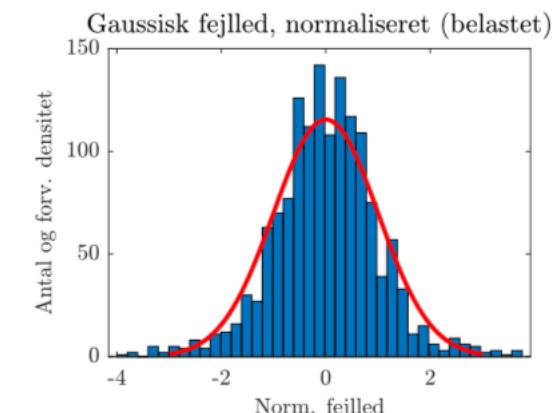
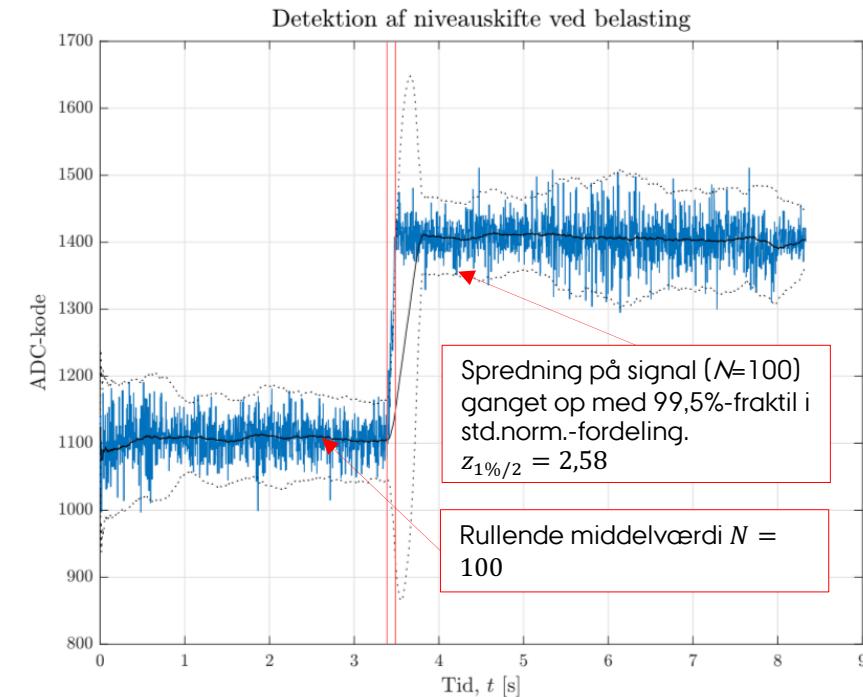
Jo flere samples, der benyttes til
middelværdi, jo lavere er variansen på
middelværdien

- > Varians på output af et MA(N)-filter

AC-middeleffekt
Gennemsnitlig kvadratafvigelse
Hvis $\bar{x} = 0$: Effekt af signalet

AC-middelamplitude
Samme fysiske enhed som x .
Hvis $\bar{x} = 0$: RMS-værdi af signalet

Illustration



OPSUMMERING

Kort om stokastiske signaler

Måling af SNR – adskillelse af signal og støj lettere i frekvensdomænet

Eksempel på betydning af SNR ved case 1: SNR for lav -> dårlig kommunikation

Mulige alternative løsninger

Lidt ekstra om stokastiske variabler

3. FIR-FILTERDESIGN

Spørgsmål 3

Diskutér FIR-filterdesign med vinduesmetoden. Herunder orden, impulsrespons, group-delay, amplitude-, og faserespons.

1. Overblik over FIR- vs. IIR-filterdesign
2. Vigtige begreber
3. Principper for design med vinduesmetoden
4. Eksempel på filterdesign med vinduesmetoden
5. Lidt ekstra om vinduer i frekvensdomænet (evt.)
6. Opsummering

FIR- VS. IIR-FILTERDESIGN

Ønsket filterrespons : $H_d(e^{j\omega})$

FIR

$$H(e^{j\omega}) = \sum_{k=0}^M b(k)e^{-j\omega k}$$

Frekvens-respons

Design og
koefficienter

Kan designe nærmest arbitraet amplitude- og faserespons

Koefficienter sættes via optimering:

- Arbitraet respons vha. minimering af forskel mellem H_d og H .
- Øger antal tappe/koefficienter, indtil approksimation er OK.

Symmetriske el. antisymmetriske koefficienter -> Lineær fase

Performance

Kan kræve højt antal tappe for at opnå ønsket performance

Lineær fase (givet ovenstående)

Konstant group delay -> ingen faseforvrængning.

Hvorfor

Simpelt design

- Arbitraet respons
- Ingen faseforvrængning
- Nemt at kvantisere (all-zero)

IIR

$$H(e^{j\omega}) = \frac{\sum_{k=0}^M b(k)e^{-j\omega k}}{\sum_{k=0}^N a(k)e^{-j\omega k}}, a_0 = 1$$

Kan approksimere gain

- Svært at approksimere fase, kontrollere fase i design

Simple filtre (fx et 2. ordens 'notch'):

- Placering af poler og nulpunkter i z-planet

Mere komplikerede filtre

- Transformation af analoge filtre
- Ingen lette metoder til at designe arbitraære responser

Kan give ønsket performance med lavere antal tappe end FIR.

Ulempen:

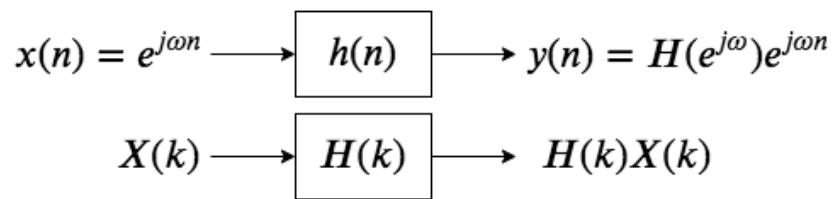
- Sjældent/aldrig lineær fase
- Group delay **IKKE** konstant, forskellige frekv., forskelligt delay
- -> Faseforvrængning

Realtids-implementering – minimér antal MAC-operationer

- Men vanskeligere/mere sensitivt at kvantisere (poler)

Hurtigere settling-time i rekursive filtre

VIGTIGE BEGREBER



Type

Frekvens-
respons for
filter

Formel

$$H(e^{j\omega}) = \sum_{k=-\infty}^{\infty} h(k)e^{-j\omega k}$$

Amplitude

$$H(e^{j\omega}) = |H(e^{j\omega})| \angle H(e^{j\omega})$$

Fase

$$\angle H(e^{j\omega}) = \Theta(\omega)$$

Lineær fase

$$\Theta(\omega) = -\alpha\omega$$

Group delay
for FIR-filter

$$D(\omega) = -\frac{d}{d\omega}\Theta(\omega) = \alpha \quad D(\omega) = \frac{N-1}{2}$$

Spec. for FIR

Kausalitet,
filtrering

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

Hvis kausale, så kun fra $k = 0$.

Betydning

Frekvensrespons er DTFT eller DFT (samplet DTFT)

- Består af amplituderespons og faserespons
- Påtrykkes en sinus, fås en faseforskudt sinus evt. med gain

Eksempel: Ideelt delay:

- Amplitderespons: 1
- Faserespons: $\theta = -\omega n_d$

Symmetriske (eller antisymmetriske) filterkoefficienter giver lineær fase

- Husk, fasen "wrapper", når vist i Matlab
- Man kan evt. også lave zero-phase filtering med *filtfilt*

Konstant group delay, hvis lineær fase.

N-1 er filterorden (antal delays)

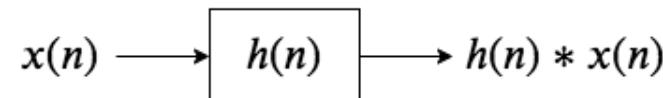
Alle frekvenser "forsinkes" lige meget gennem filteret

- Group delay kan ses som gennemsnitligt filter-delay

Alle praktiske filtre til realtid er kausale

- Kan ikke bruge samples fra fremtiden!

Så $h(k) = 0$ for $k < 0$. Kan flytte koefficienter med shift.



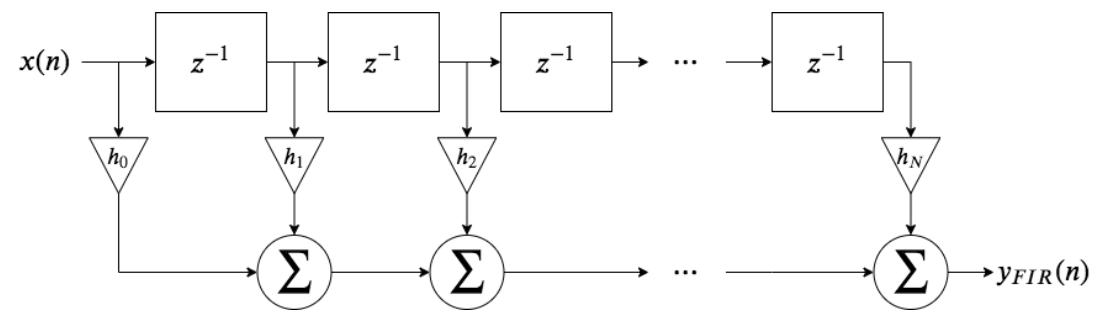
DESIGN MED VINDUESMETODEN

Overordnede principper

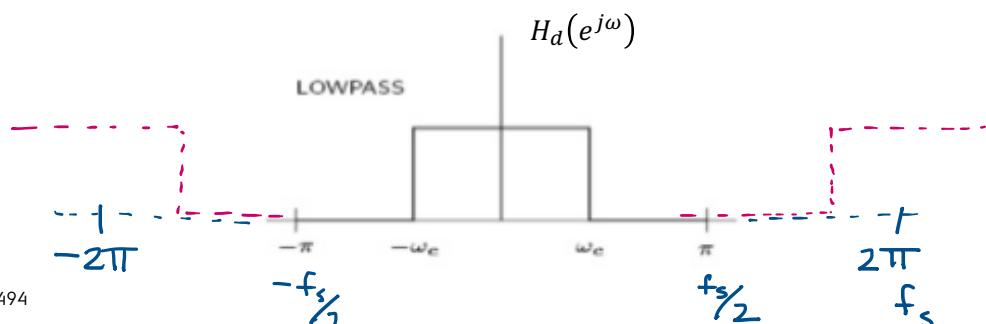
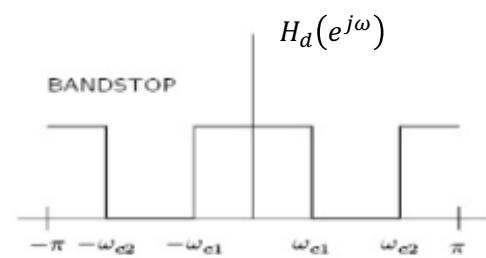
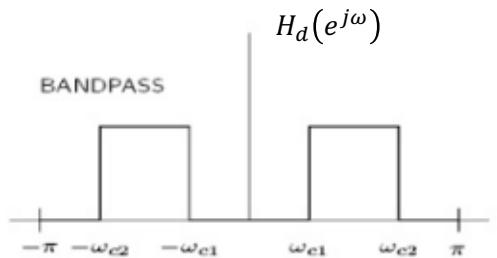
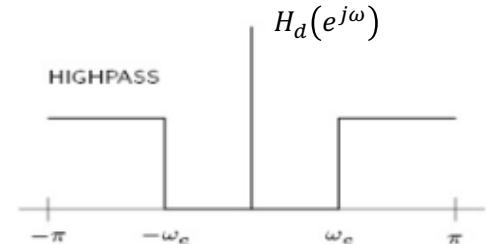
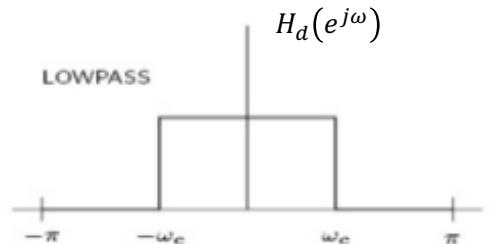
1. Design ønsket frekvensrespons, $H_d(e^{j\omega})$ eller $H_d(k)$, i frekv.dom.
 - Diskret frekvensrespons, $H_d(k)$ designes
 - Algebraisk udtryk
 - Sætte værdier af frekvenssamples i $H(k)$
 - Husk periodicitet med f_s (2π)!
2. Transformér til filterkoefficienter / impulsrespons
 - IDFT giver impulsrespons som for FIR-filter = filterkoeff.
 - IDFT af algebraisk udtryk eller IFFT af frekvenssamples
 - IFFT er nemmest ☺
3. Bekræft frekvensrepsons. Benyt vinduer til at dæmpe ripples
 - Vindue i tidsdomæne dæmper ripples i frekv.dom.

Eksperimentér med filterorden for at opnå komme "tæt nok" på ønsket respons og samtidig overholde maks. group delay, indsvingning osv.

(husk: længere impulsrespons, tættere på ideelt)



Frekvensrespons for idelle digitale filtre



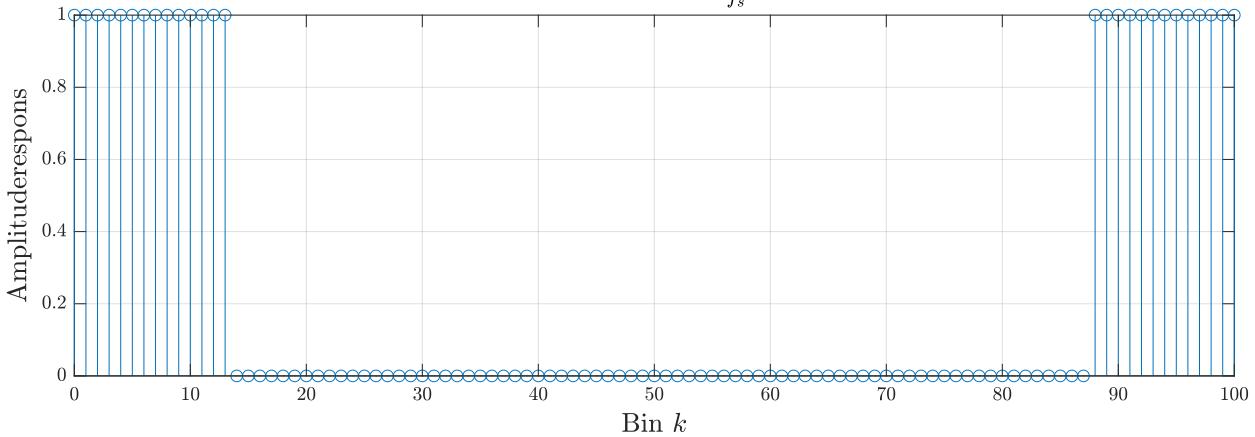
DESIGNEKSEMPEL: LAVPASFILTER

Step 1. Design amplituderespons

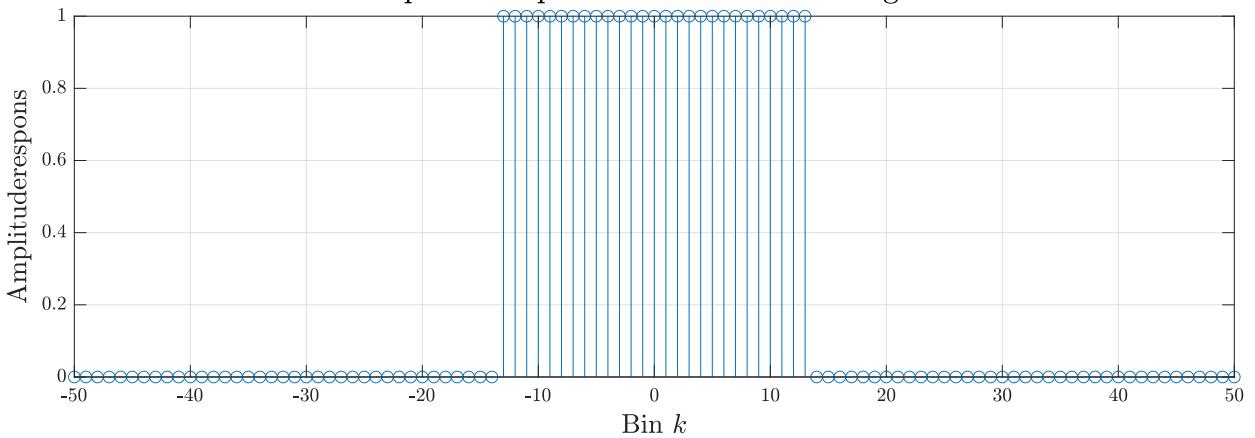
- Gain sættes til 1 i pasbånd, 0 i stopbånd
- Spejlet / periodisk

Også vist med symmetri hvor "lave frekvenser" er omkring 0.

Ideelt amplituderespons, H_d
 $M = 100$ (orden), $N = 101$ (filterkoeff.), $f_s = 8192$ Hz
 $f_c = 1024$ Hz, $k_c = \frac{f_c}{f_s}N = 13$



Amplituderespons: Shiftet som vist i figurer

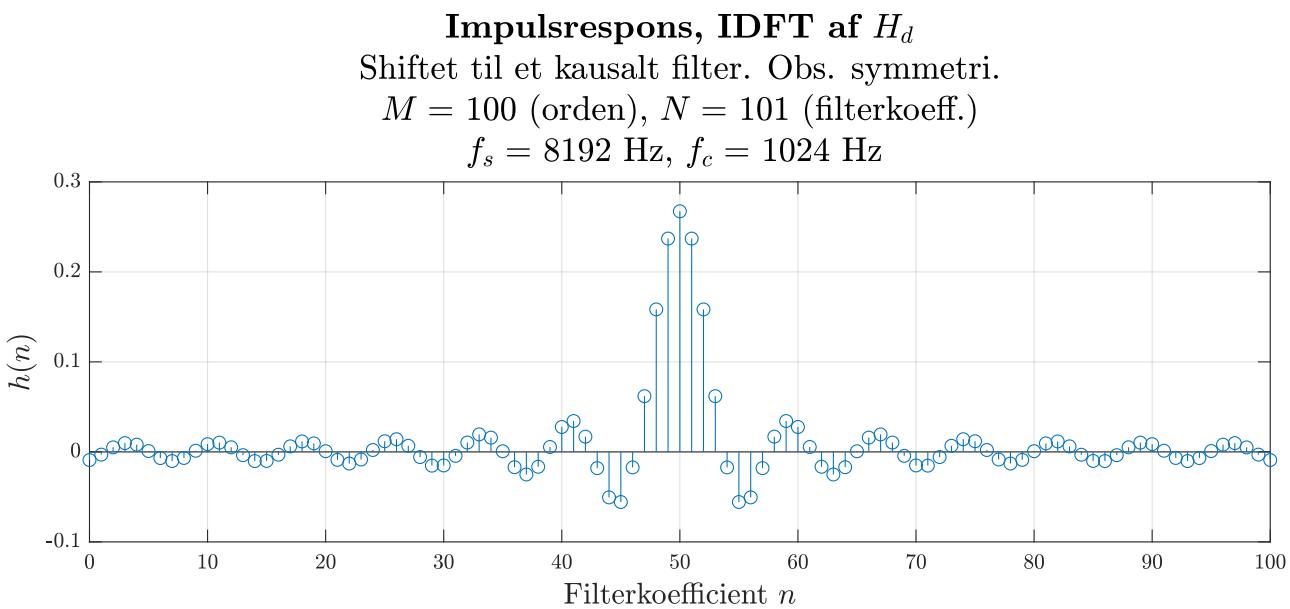


LPF FORTSAT

Step 2. Transformér til impulsrespons

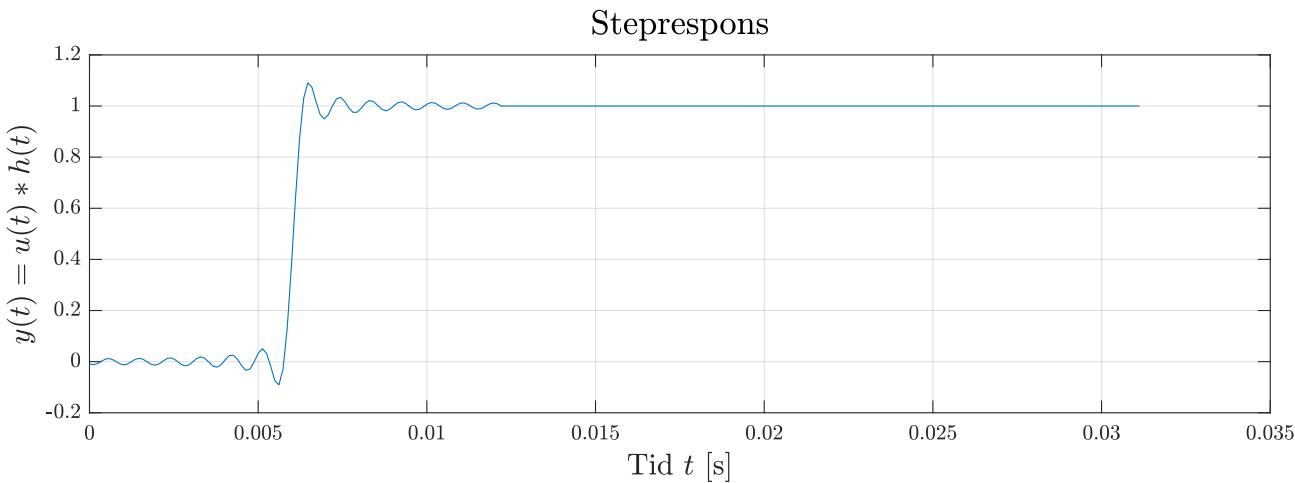
- Invers transformation

$$h(k) = IDFT\{H_d(k)\}$$



Bemærk:

- Sinc-funktion (modsvarer "firkant" i frekv.dom.)
- Ulige antal koefficienter -> symmetri
- Symmetriske koefficienter, så forventer lineær fase (konstant group delay)
- Steprespons udviser ringninger (Gibb's)



LPF FORTSAT

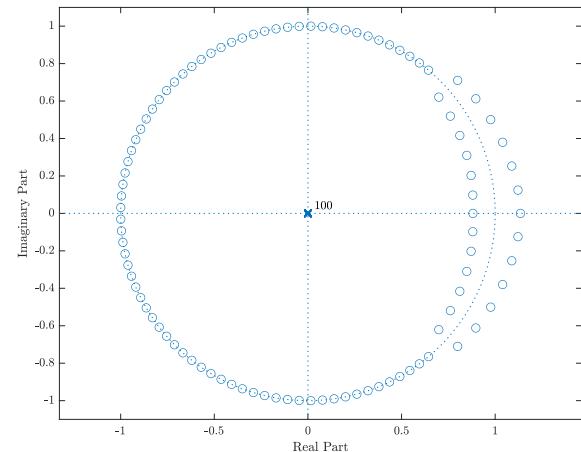
Step 3. Faktisk frekvensrespons

Filteret har respons som forventet

- Knækfrekvens omkring f_c
 - Stejlere filter, smallere transitionsbånd -> øg N
- MEN:
 - Ripples i pasbånd og stopbånd (Gibb's)
 - Kan fornemmes fra nulpunkter i z-planet
 - -> benyt vinduer
- Lineær fase i pasbåndet

Group delay konstant = 50.

- Bekræftes med hældning på fase omkring -50 og Matlabs grpdelay(h)

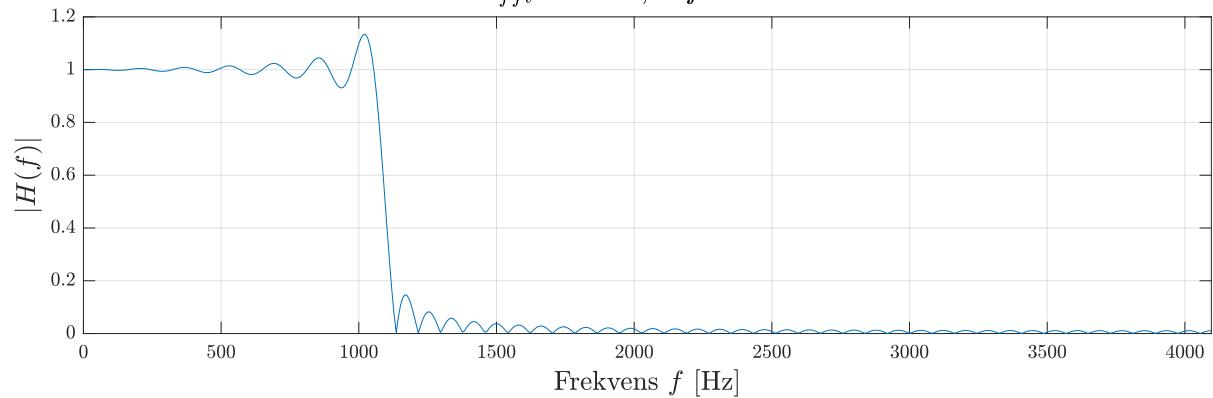


Faktisk amplituderespons, FFT af $h(n)$

$M = 100$ (orden), $N = 101$ (filterkoeff.)

$f_s = 8192$ Hz, $f_c = 1024$ Hz

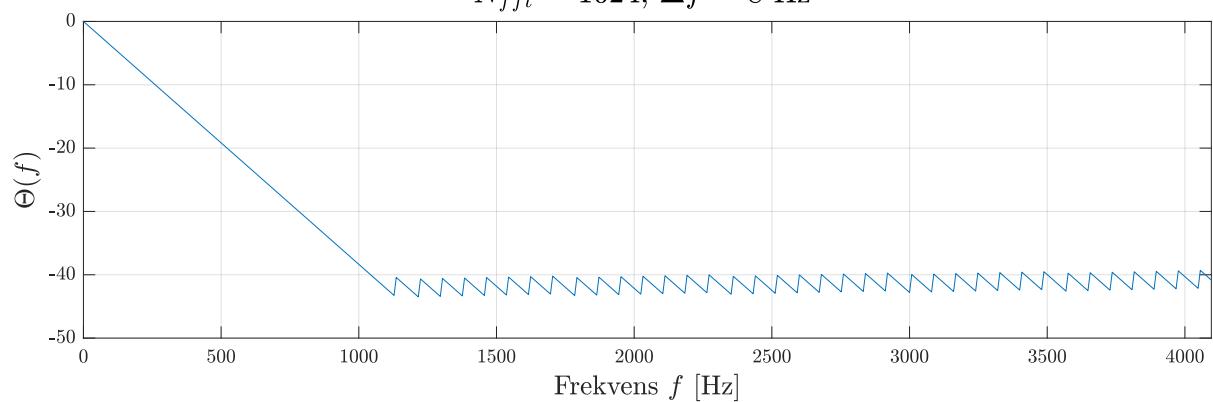
$N_{fft} = 1024$, $\Delta f = 8$ Hz



Faktisk faserespons

Beregnet med *unwrap*

$N_{fft} = 1024$, $\Delta f = 8$ Hz



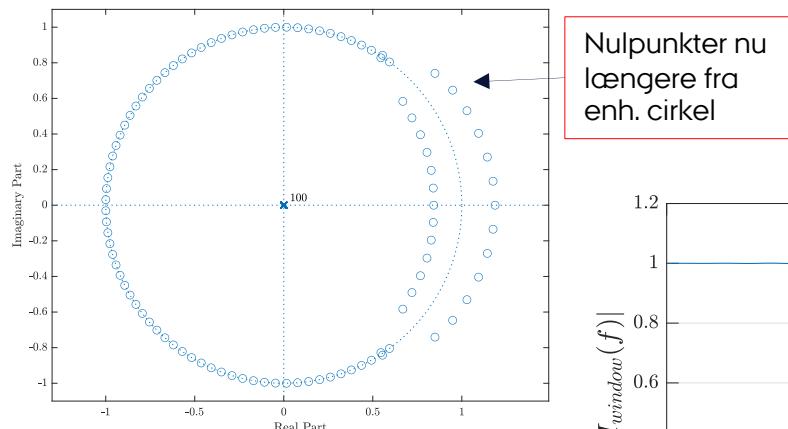
Bemærk: Kausalt FIR-filter med N -poler i nulpunktet.

LPF MED VINDUER

Step 3b. Tilføj vindue

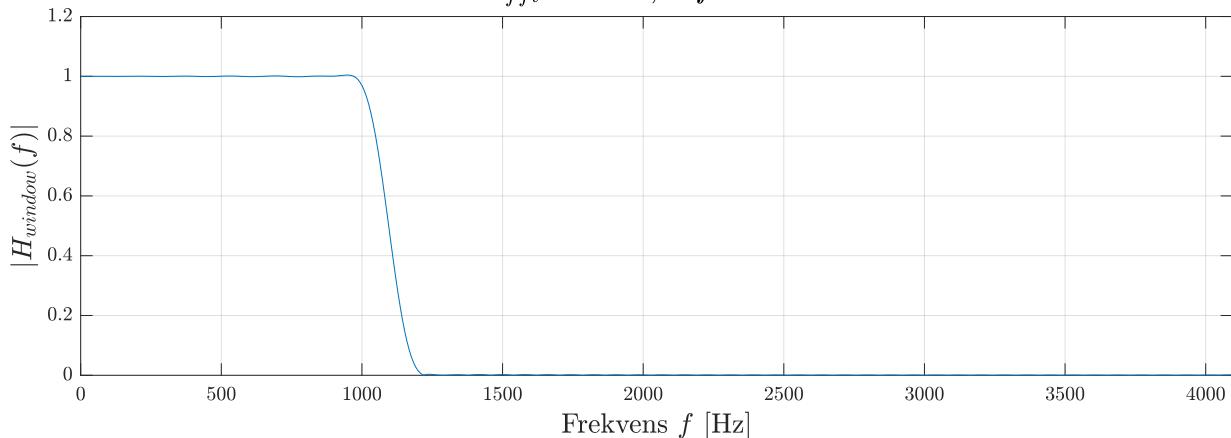
Benytter et Hamming-vindue

- Bredere transitionsbånd
- -41 dB dæmpning af side lobes



Nulpunkter nu længere fra enh. cirkel

Amplituderespons, FFT af $w_{hamming}(n) \cdot h(n)$
 $M = 100$ (orden), $N = 101$ (filterkoeff.)
 $f_s = 8192$ Hz, $f_c = 1024$ Hz
 $N_{fft} = 1024$, $\Delta f = 8$ Hz



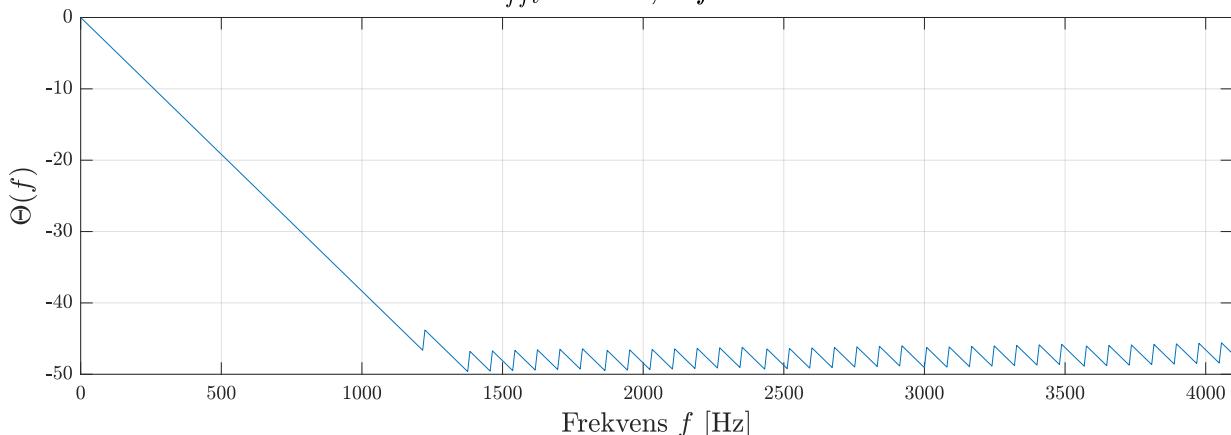
Pænere respons

- Ripples dæmpes af vindue, kan evt. "fikses" ved at øge orden

Vigtige observationer, man kan gøre sig:

- Orden -> skarphed af filter
 - Antal filterkoefficienter påvirker ikke ripple peak (kun hastighed på ripples)
- Ripples - pga. rektangulært vindue (Gibbs)
 - Dæmpes med vinduesfkt.!
- Ripples skyldes Gibbs fænomen.
 - Repræsenterer en firkant fra en sum af sinusfkt.

Faserespons efter vindue
Beregnet med *unwrap*
 $N_{fft} = 1024$, $\Delta f = 8$ Hz



VINDUER OG FOLDNINGSSÆTNINGEN

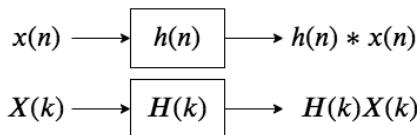
Hovedresultat

For LTI-systemer; DTFT, DFT (og FFT), z-transformation, mv.

Foldning i tidsdom. \leftrightarrow multiplikation i frekvensdom.

$$h(n) * x(n) \xrightarrow{DFT} H(k) \cdot X(k)$$

Anvendelse: Effektiv filtrering vha. DFT/FFT

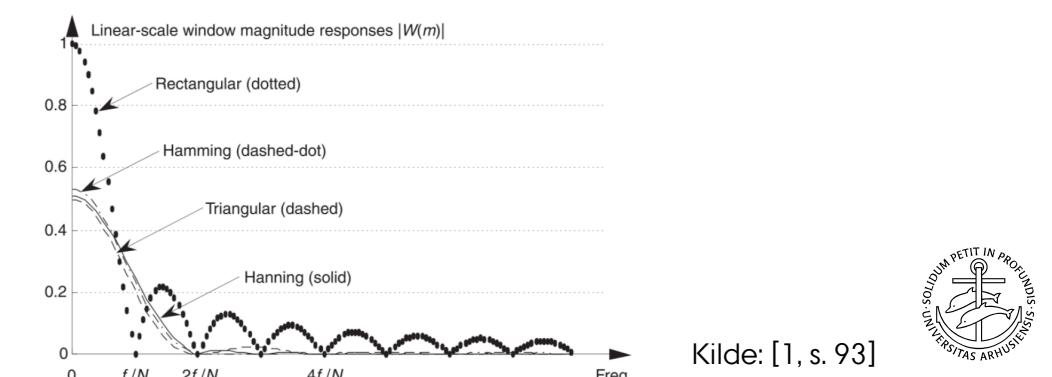
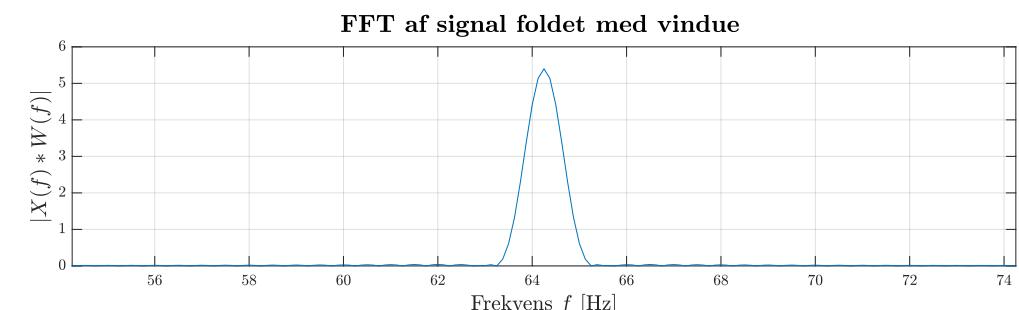
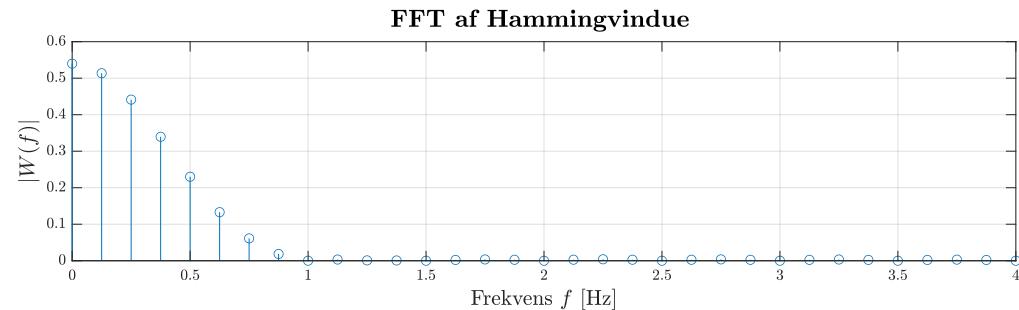
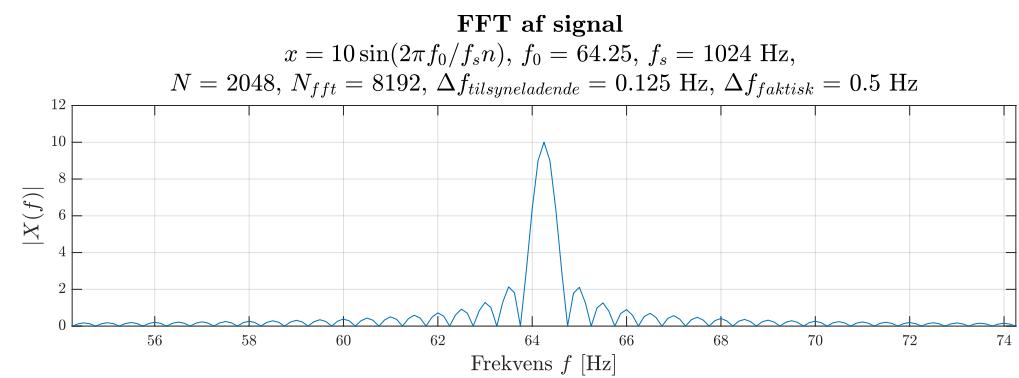


Multiplikation i tidsdom. \leftrightarrow foldning i frekvensdom.

$$h(n) \cdot x(n) \xrightarrow{DFT} H(k) * X(k)$$

Anvendelser:

- Vinduer \rightarrow foldning med smallt filter i frekvensdom.
- Modulation, fx AM. "Frekvensshifting".



OPSUMMERING

FIR-design vs. IIR-design

Vigtige begreber og designparametre

Designeksempel

Et par vigtige observationer, man kan gøre sig

4. IIR-FILTRE

Spørgsmål 4

Diskutér IIR-filterdesign med pol/nulpunktsplacering og eventuelt bilineær z-transformation. Referér til Case-projekt 2.

1. IIR vs. FIR – hvorfor IIR?
2. Frekvensrespons og z-transformation
3. Eksempel på design vha. pol-nulpunktsplacering (case 2)
4. Anvendelser og implementering på fixed-point platform (case 2)
5. Design med BZT
6. Opsummering

FIR- VS. IIR-FILTERDESIGN

Ønsket filterrespons : $H_d(e^{j\omega})$

FIR

$$H(e^{j\omega}) = \sum_{k=0}^M b(k)e^{-j\omega k}$$

Frekvens-respons

Design og
koefficienter

Kan designe nærmest arbitraet amplitude- og faserespons

Koefficienter sættes via optimering:

- Arbitraet respons vha. minimering af forskel mellem H_d og H .
- Øger antal tappe/koefficienter, indtil approksimation er OK.

Symmetriske el. antisymmetriske koefficienter -> Lineær fase

Performance

Kan kræve højt antal tappe for at opnå ønsket performance

Lineær fase (givet ovenstående)

Konstant group delay -> ingen faseforvrængning.

Hvorfor

Simpelt design

- Arbitraet respons
- Ingen faseforvrængning
- Nemt at kvantisere (all-zero)

IIR

$$H(e^{j\omega}) = \frac{\sum_{k=0}^M b(k)e^{-j\omega k}}{\sum_{k=0}^N a(k)e^{-j\omega k}}, a_0 = 1$$

Kan approksimere gain

- Svært at approksimere fase, kontrollere fase i design

Simple filtre (fx et 2. ordens 'notch'):

- Placering af poler og nulpunkter i z-planet

Mere komplikerede filtre

- Transformation af analoge filtre
- Ingen lette metoder til at designe arbitraære responser

Kan give ønsket performance med lavere antal tappe end FIR.

Ulempler:

- Sjældent/aldrig lineær fase
- Group delay **IKKE** konstant, forskellige frekv., forskelligt delay
- -> Faseforvrængning

Realtids-implementering – minimér antal MAC-operationer

- Men vanskeligere/mere sensitivt at kvantisere (poler)

Hurtigere settling-time i rekursive filtre

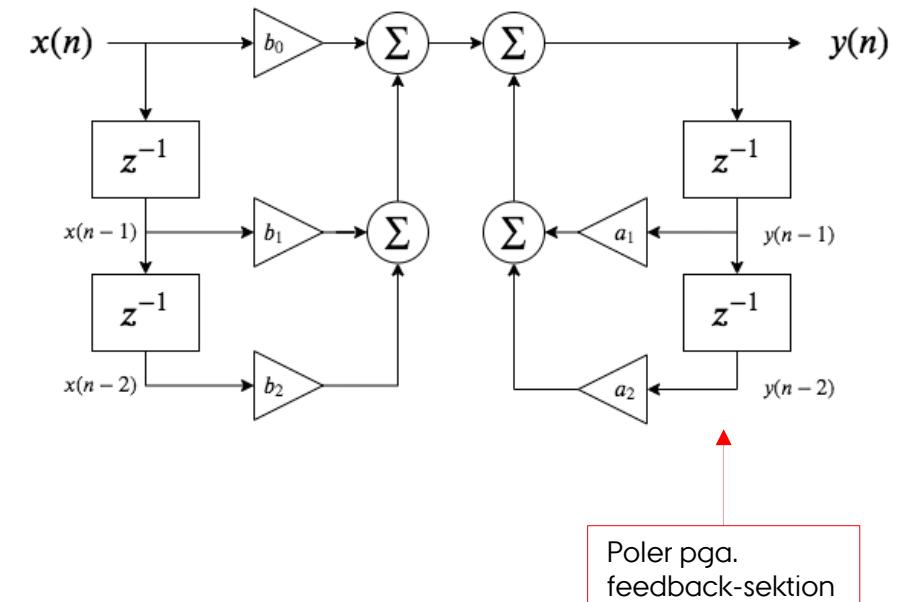
IIR-FILTER HAR FEEDBACK-SEKTION

Differensligning og signalgraf

$$y(n) = b_0x(n) + b_1x(n-1) + b_2x(n-2) + a_1y(n-1) + a_2y(n-2)$$

Generel differensligning

$$y(n) = \sum_{m=0}^M b_m x(n-m) + \sum_{k=1}^K a_k y(n-k)$$



Z-TRANSFORMATIONEN BENYTTES TIL AT "LØSE" DIFFERENSLIGNING

$$y(n) = \sum_{m=0}^M b_m x(n-m) + \sum_{k=1}^K a_k y(n-k)$$

$$x(n) \longleftrightarrow X(z)$$

Z-transformation til at "løse" differensligning, ovf. fkt.

$$x(n-m) \longleftrightarrow z^{-m}X(z)$$

$$Y(z) = X(z) \sum_{m=0}^M b_m z^{-m} + Y(z) \sum_{k=1}^K a_k z^{-k}$$

Bemærk fortegn ift.
differensligning

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{m=0}^M b_m z^{-m}}{1 - \sum_{k=1}^K a_k z^{-k}}$$

FREKVENSRESPONS ER EN TUR PÅ ENH.CIRK.

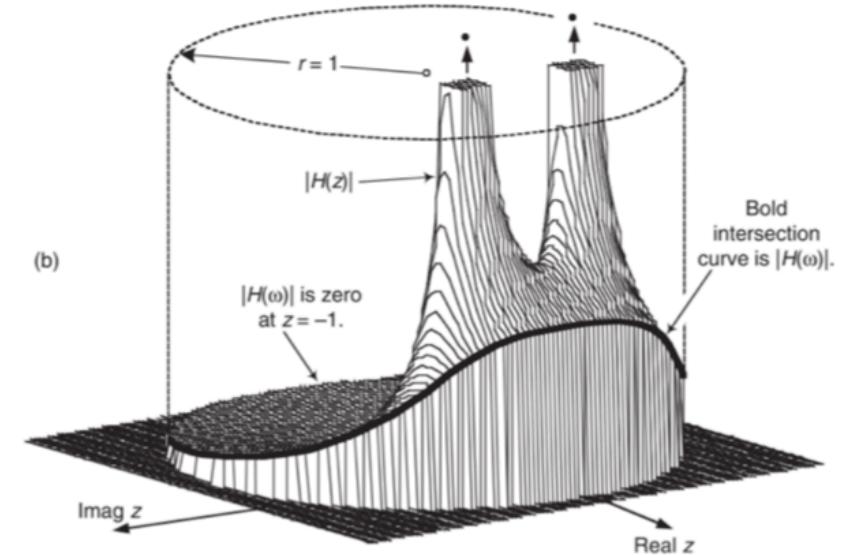
$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{m=0}^M b_m z^{-m}}{1 - \sum_{k=1}^K a_k z^{-k}}$$

Fouriertransformation er frekvensrespons, variabelsubst.

$$z = e^{j\omega} \longrightarrow H(e^{j\omega}) = \frac{Y(e^{j\omega})}{X(e^{j\omega})} = \frac{\sum_{m=0}^M b_m e^{-j\omega m}}{1 - \sum_{k=1}^K a_k e^{-j\omega k}}$$

På enhedscirkel

- Ratio af to polynomier (DTFT'er)
- Nulpunkter og poler
- Regnes numerisk som ratio af to FFT'er med samme længde

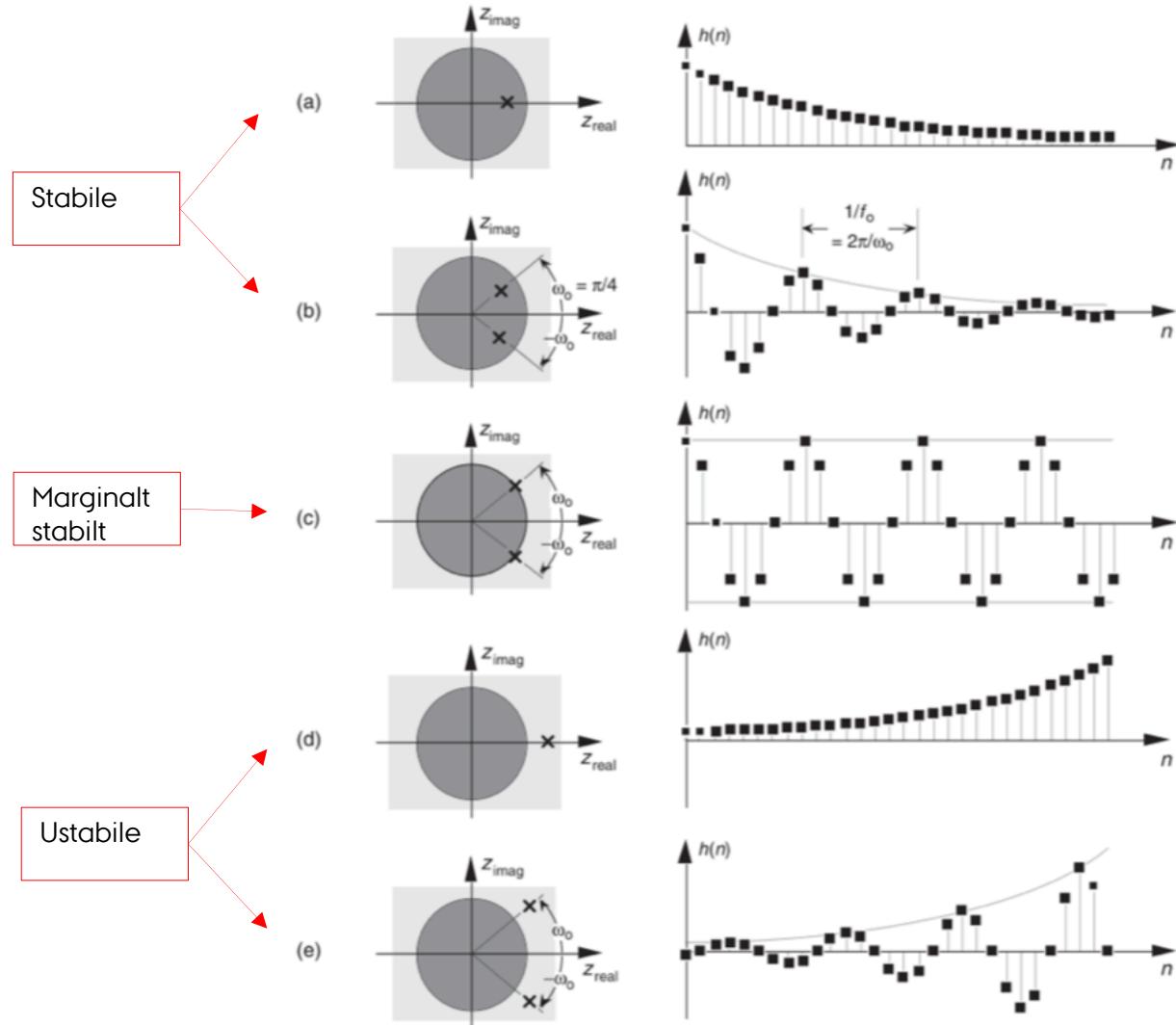


Kilde: [1, s. 283]

POLPLACERING I Z-PLANET

Placering af poler

- Stabilt filter:
 - Poler inden for enhedscirkel
- Marginalt stabilt (oscillerende):
 - Poler på enhedscirkel
- Ustabile
 - Poler uden for enhedscirkel



Kilde: [1, s. 273]

CASE 2: NOTCH-FILTER

Centerfrekvensen for notch-filteret er ω_c [rad/s] med $\omega_c = \pi \frac{f_c}{f_s/2}$

Pol-nulpunktsparrene er komplekst konjugerede, fås følgende nulpunkter og poler: $z_0 = e^{j\omega_c}$, $z_1 = e^{-j\omega_c}$, $p_0 = re^{j\omega_c}$ og $p_1 = re^{-j\omega_c}$.

Filterets overføringsfunktion findes i den faktoriserede form:

$$H(z) = \frac{(z - z_0)(z - z_1)}{(z - p_0)(z - p_1)} = \frac{(z - e^{j\omega_c})(z - e^{-j\omega_c})}{(z - re^{j\omega_c})(z - re^{-j\omega_c})} \quad [1, 6-38 s. 285]$$

Ganges paranteserne ud og benyttes Eulers relation $2 \cos(\omega) = e^{j\omega} + e^{-j\omega}$, s har vi:

$$H(z) = \frac{z^2 - 2 \cos(\omega_c)z + 1}{z^2 - 2r \cos(\omega_c)z + r^2}$$

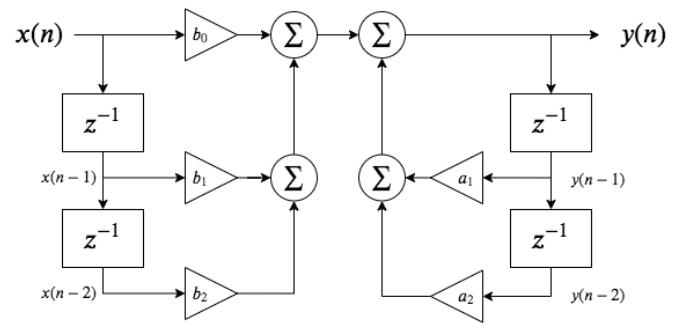
Brug gain-faktor G til at normalisere responset til et gain på 1 i pasbndet. Forkort z^{-2} for at se systemfunktionen:

$$\frac{Y(z)}{X(z)} = G \cdot H(z) = G \frac{1 - 2 \cos(\omega_c)z^{-1} + z^{-2}}{1 - 2r \cos(\omega_c)z^{-1} + r^2 z^{-2}} \equiv \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 - a_1 z^{-1} - a_2 z^{-2}} = \frac{B(z)}{A(z)}$$

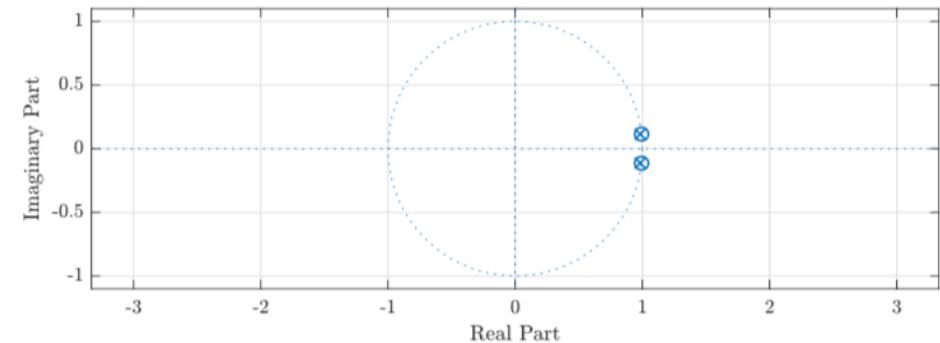
Ved sammenligning ses, at $b_0 = b_2 = G$, $b_1 = -2G \cos(\omega_c)$, $a_0 = 1$ (altid!), $a_1 = 2r \cos(\omega_c)$ og $a_2 = -r^2$.

Case: Ønsker at fjerne forstyrrende tone ved 876 Hz.

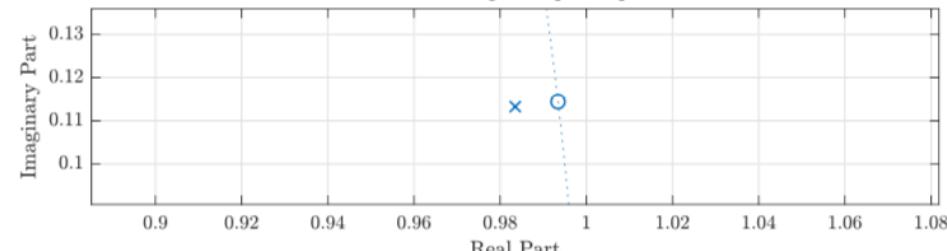
- Placere poler og nulpunkter, vælge vinkel svarende til 876 Hz
- Implementere på fixed-point-platform (Texas DSP)
- Jo tættere på hinanden, jo skarpere notch vs. kvantisering.



Pol-nulpunktsplacering for notch-filter



Udsnit: Et pol-nulpunktspar



CASE 2: NOTCH-FILTER

```

1 r = 0.99; % Modulus for poler ("Q-faktor")
2 wc = pi * round(fidx) / (fs / 2); % Centerfrekvens [rad/s]
3 K = 1; % Til første iteration af DC gain
4
5 for iteration=1:2
6 G = 1/K; % Skalering så DC Gain = 1 (0 dB). G=0.9991.
7
8 % Følgende koefficienter ift. differensligning.
9 b0 = G; % z^0
10 b1 = -2*cos(wc)*G; % z^-1
11 b2 = G; % z^-2
12 a1 = 2*r*cos(wc); % z^-1
13 a2 = -r*r; % z^-2
14
15 % Følgende polynomier ift. overføringsfkt.
16 b = [b0 b1 b2]; % B(z)
17 a = [1 -a1 -a2]; % A(z)
18
19 K = sum(b)/sum(a); % DC gain (Lyons s. 300)
20 end

```

```

1 % Differensligning:
2 feedforward = [num2str(b0) ' x(n) ' ...
3 num2str(b1) ' x(n-1) ' ...
4 num2str(b2) ' x(n-2) '];
5 feedback = [num2str(a1) ' y(n-1) ' ...
6 num2str(a2) ' y(n-2) '];
7 diffeq = ['y(n) = ' feedforward ' + ' feedback];
8 disp(diffeq);
9
10 Hsys = tf(b, a, 1/fs) % Vis fin overføringsfunktion

```

$y(n) = 0.99761 x(n) - 1.9821 x(n-1) + 0.99761 x(n-2) + 1.967 y(n-1) - 0.9801 y(n-2)$
 $H_{sys} =$
 $0.9976 z^2 - 1.982 z + 0.9976$
 $-----$
 $z^2 - 1.967 z + 0.9801$

Vælg r for at få skarpt notch
Beregn vinkel i z-plan.

Iterativt. Sikrer gain = 1 i
pasbånd

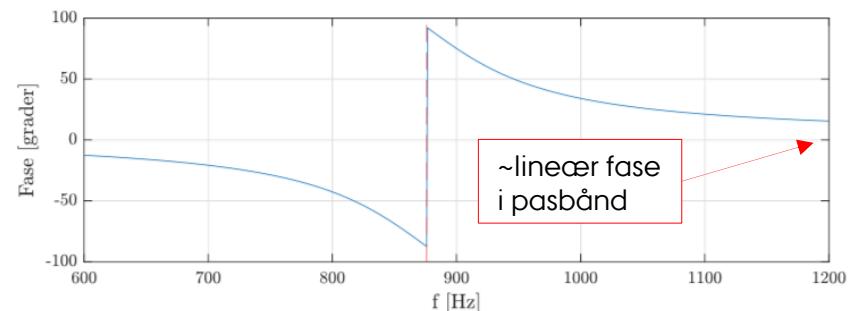
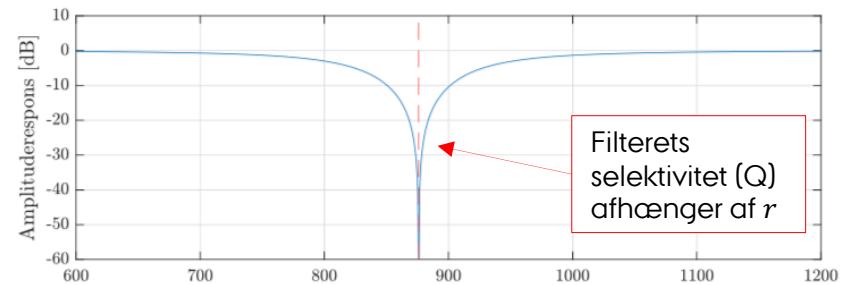
Beregn koefficienter som
udledt på tidl. slide

Polynomier til
overføringsfunktion

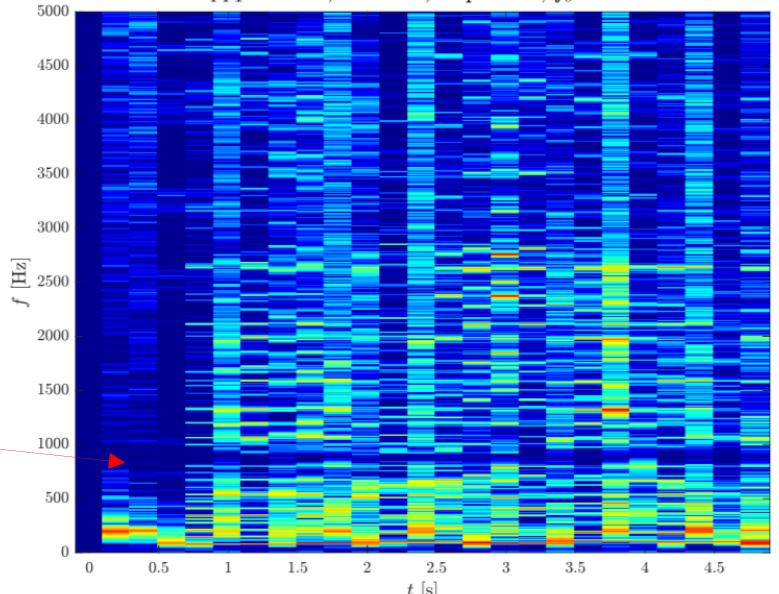
Vis differensligning og
overføringsfunktion

Tone væk, filter virker
Men har fjernet noget
brugbart signal (adaptiv filt.?)

Designet notch-filter



Spektrogram
 $N_{FFT} = 16384, L = 9600, \text{step} = 9600, f_s = 48000$



CASE 2: IMPLEMENTERING PÅ DSP

Kvantisering af koefficienter til 16-bit fixed-point DSP

- S15-format -> Nogle koeff. for store
- Manuelt eller vha. Matlab filterDesigner

```
1 K = 2^15; % Benyttes til <<15 og >>15 operationer
2
3 % Koefficienter i S15
4 b0_ = round(b0 * K); % Svarer til (short) (b0 << 15)
5 b1_ = round(b1/2 * K); % Bem. b1/2, så skal akkumuleres dobbelt
6 a1_ = round(a1/2 * K); % Bem. a1/2, så skal akkumuleres dobbelt
7 a2_ = round(a2 * K);
```

Værdiområdet for S15 er $1 \leq x < 1 - 2^{-15}$

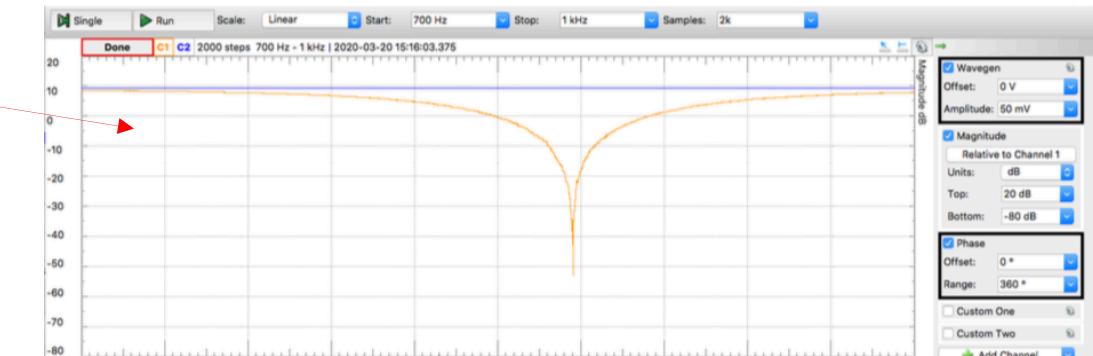
I S15 *forestiller* vi os, at der er et binærkomma $k = 15$ pladser fra højre (LSB), og at MSB repræsenterer tallets fortegn. S15 bitmønsteret med indsats binærkomma er $b_0.b_1b_2b_3 \dots b_{13}b_{14}b_{15}$. Mønsteret repræsenterer radix-10 kommatalsværdien $-b_0 + \sum_{n=1}^{k=15} b_n 2^{-n}$.

Implementering af differensligning

```
15 /* difference equation, coerce all data into
   * sign extended 32-bit words during calculation */
16 acc = ( (long) b[0] * input ); /* b0 x(n) */
17 acc += ( (long) b[1] * dx[0] ); /* b1 x(n-1) */
18 acc += ( (long) b[1] * dx[0] ); /* added twice due to scaling */
19 acc += ( (long) b[2] * dx[1] ); /* b2 x(n-2) */
20 acc += ( (long) a[1] * dy[0] ); /* a1 y(n-1) */
21 acc += ( (long) a[1] * dy[0] ); /* added twice due to scaling */
22 acc += ( (long) a[2] * dy[1] ); /* a2 y(n-2) */
23
24
25 /* Update delay line */
26 dx[1] = dx[0]; /* x(n-2) = x(n-1) */
27 dx[0] = input; /* x(n-1) = x(n) */
28 dy[1] = dy[0]; /* y(n-2) = y(n-1) */
29 dy[0] = (short) acc; /* y(n-1) = y(n) */
```

Tjek filterkarakteristik

- Filter skal ikke blive ustabilt
- Kvalitativ vurdering af lydkvalitet
- Dæmper omkring 40 dB i centerfrekvens



BILINEAR Z-TRANSFORM

Idé:

Approksimér overføringsfunktion for et analogt prototype-filter med ditto fra et diskret filter

- Analogt filter $H_c(s)$
- Diskret filter $H(z)$

Mapping til digital

$$H(z) = H_c(s)|_{s=\frac{2}{T_s} \frac{z-1}{z+1}}$$

Tilgang:

Mapper venstre s-halvplan til enhedscirkel i z-plan

- Uendelig imaginær akse i s-plan mappes ind på enhedscirkel
 - Ikke-lineær sammenhæng mel. frekvenser -> warping
 - Man kan evt. pre-warpe omkring den kritiske frekvens

Analog frekvens

$$\omega_a = \frac{2}{T_s} \tan\left(\frac{\omega_d}{2}\right)$$

Digital frekvens

$$\omega_d = 2 \tan^{-1}\left(\frac{\omega_a T_s}{2}\right)$$

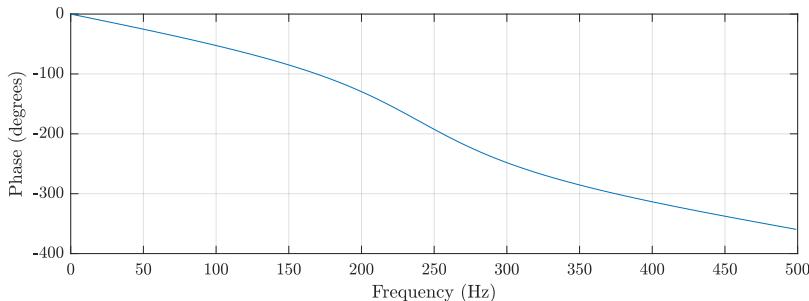
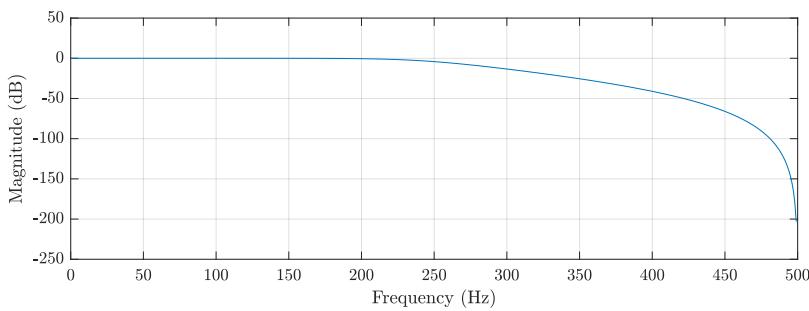
Højre halvplan (ustabilt) mappes uden for enhedscirkel.

4. ORDENS IIR BUTTERWORTH IIR VIA BZT MAXIMALLY FLAT

Via bilinear z-transform

```
fs = 1000; % Samplingsfrekvens
fc = 300; % Analog knækfrekvens, Hz
n = 4; % 4. ordens analogt filter
[B,A] = butter(n, 2*pi*fc, 's'); % Design analogt filter
[bz,az] = bilinear(B, A, fs); % BZT

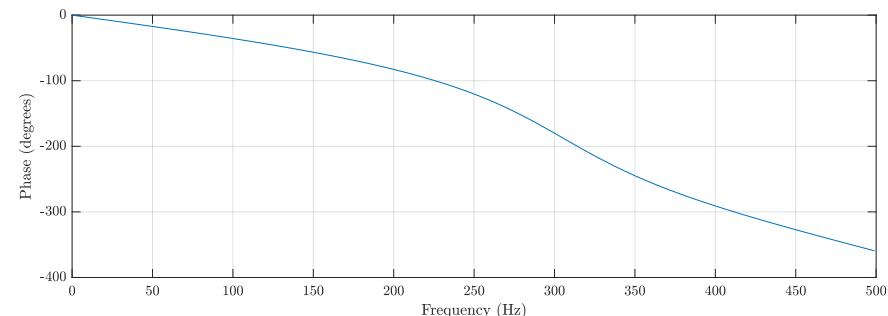
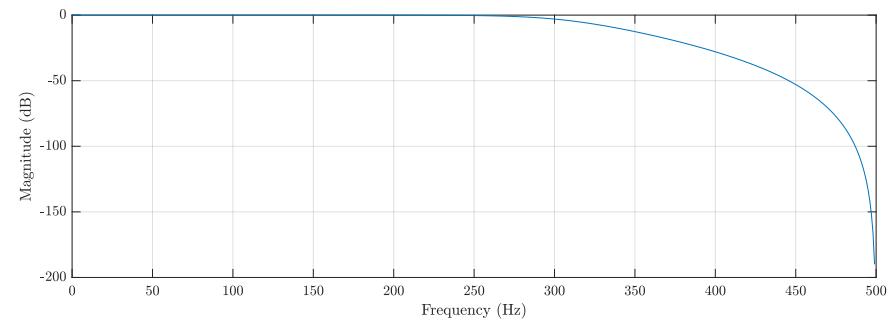
figure;
freqz(bz, az, 512, fs); % Analyse med 512 samples
```



"Direkte" som digital filter

```
fs = 1000;
fc = 300;

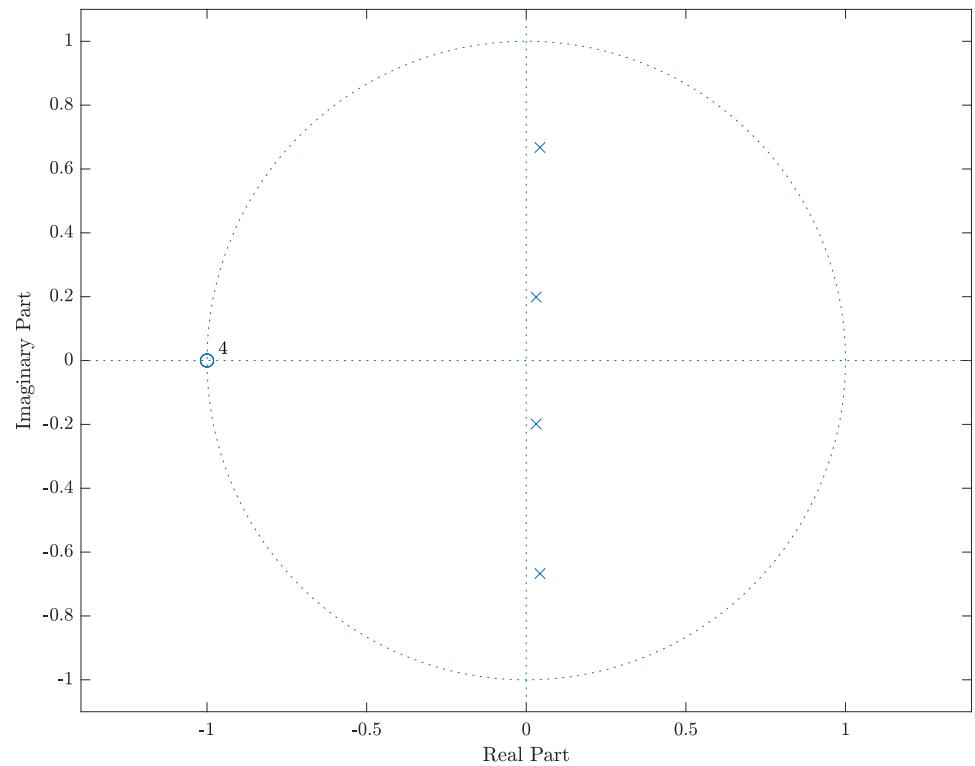
[b,a] = butter(4,fc/(fs/2));
figure;
freqz(b,a,512,fs)
```



4. ORDENS BUTTEWORTH IIR

Man kan forestille sig en tur rundt på enhedscirkel

- Stemmer (selvfølgelig) overens med frekvensrespons



OPSUMMERING

IIR vs. FIR

Simpelt filter – placering af poler og nulpunkter

Implementering på fixed-point platform

BZT for mere avancerede filtre

Perspektivering:

- AR-model: IIR all-pole. Fx forecasting med AR-proces, eller fitting til parametrisk PSD
- ARMA-model kombinatio med feedforward og feedback

5. STOKASTISKE SIGNALER OG MIDLING

1. Overblik, stokastiske signaler
2. Statistiske mål
3. Midlingsfiltre
 1. MA, MA-rekursivt
 2. Eksponentielt midlingsfilter
4. Opsummering

Spørgsmål 5

Diskutér stokastiske signaler og midlingsfiltre. Herunder middelværdi og varians. Referér til Case-projekt 3.

STOKASTISKE SIGNALER

Signaler med stokastiske komponenter. Stokastisk: Som ikke kan forudsiges præcis, muligvis med en sandsynlighedsfordeling. Def. af støj: Uønsket fluktuation i data. Fx additiv støj, hvor observerbar data = signal + støj:

$$x(n) = \mu(n) + \epsilon(n)$$

Støjen kan fx være givet ved $\epsilon \sim N(0, \sigma^2)$.

Type

Tidsdomæne

Stationære
signaler

Tids-asyriskron midling i tidsdomæne reducerer støj på estimat af "sand" middelværdi

- MA-filtre, eksponentielle midlingsfiltre

Ikke-
stationære
signaler

Tids-synkron sampling (med faseinformation)

- Parallel-midling

Opdele signal i stationære segmenter

Motivation:

- Sensordata med støj (støjgulv)
- Målinger med måleusikkerhed
- Osv.

Spektralanalyse

Power Spectral Density med reduceret støj:

- Ikke-parametrisk med fx Welch's metode
 - Midling af adskillige FFT'er dæmper overlejret støj
- Parametrisk med fx en LS-fitted AR-model
 - PSD regnes pba. a -koefficienterne.

Model så proces gøres stationær
Spektrogram

DESKRIPTIV STATISTIK

Type

Formel

Middelværdi

$$\bar{x} = \frac{\sum_{n=0}^{N-1} x(n)}{N}$$

Varians på
middelværdi

$$Var(\bar{x}) = \sigma_{\text{output}}^2 = \frac{\sigma_{\text{input}}^2}{N}$$

Varians

$$\hat{\sigma}^2 = \frac{\sum_{n=0}^{N-1} (x(n) - \bar{x})^2}{N}$$

Standard-
afvigelse
(spredning)

$$\hat{\sigma} = \sqrt{\hat{\sigma}^2}$$

Betydning

DC-niveau eller DC-offset
Estimat på den sande middelværdi i en
dataserie

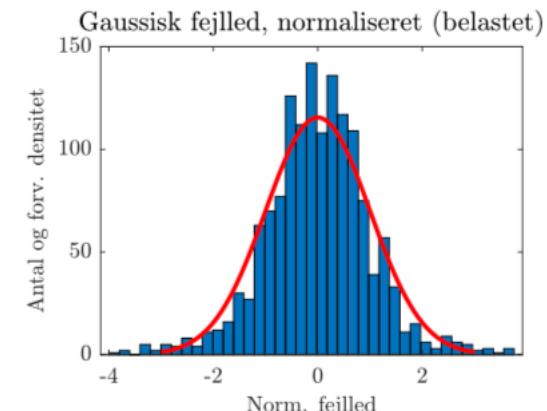
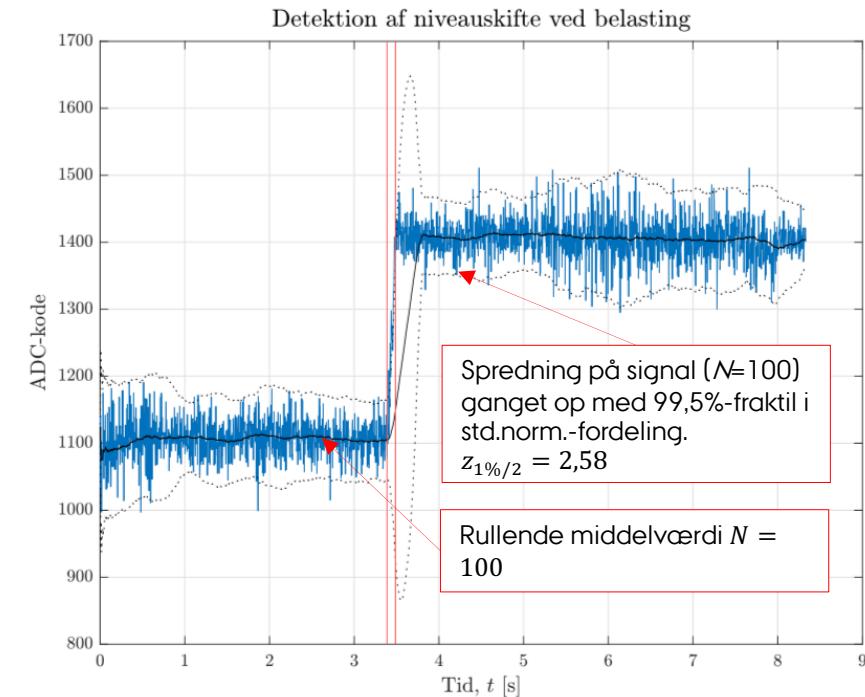
Jo flere samples, der benyttes til
middelværdi, jo lavere er variansen på
middelværdien

- > Varians på output af et MA(N)-filter

AC-middeleffekt
Gennemsnitlig kvadratafvigelse
Hvis $\bar{x} = 0$: Effekt af signalet

AC-middelamplitude
Samme fysiske enhed som x .
Hvis $\bar{x} = 0$: RMS-værdi af signalet

Illustration



OVERBLIK OVER MIDLINGSFILTRE

Typer

Differensligning

FIR MA
(ikke-rekursiv)

$$y_{MA(N)}(n) = \frac{1}{N} \sum_{k=0}^{N-1} x(n-k)$$

FIR MA
(rekursiv impl.)

$$y(n) = \frac{1}{N} (x(n) - x(n-N)) + y(n-1)$$

Eksponentielt
midlingsfilter
(IIR)

$$y(n) = \alpha x(n) + (1 - \alpha)y(n-1)$$
$$0 < \alpha < 1$$

Design

Højere N:

- Bedre midling og støjreduktion (lavere outputvarians)
- MEN længere indsvingning, transientrespons $N - 1$ samples
- MEN flere memory-elementer krævet i buffer

Samme performance som ovenstående

- Færre MAC-operationer

Færre beregninger og memory-elementer end MA-filtre

Designparameter er α

- α tæt på 1: Hurtig indsvingning
- α tæt på 0: "lang" midling
- Kan varieres "undervejs"

SAMMENLIGNING AF MA-FILTRE

Som på sidste slide:

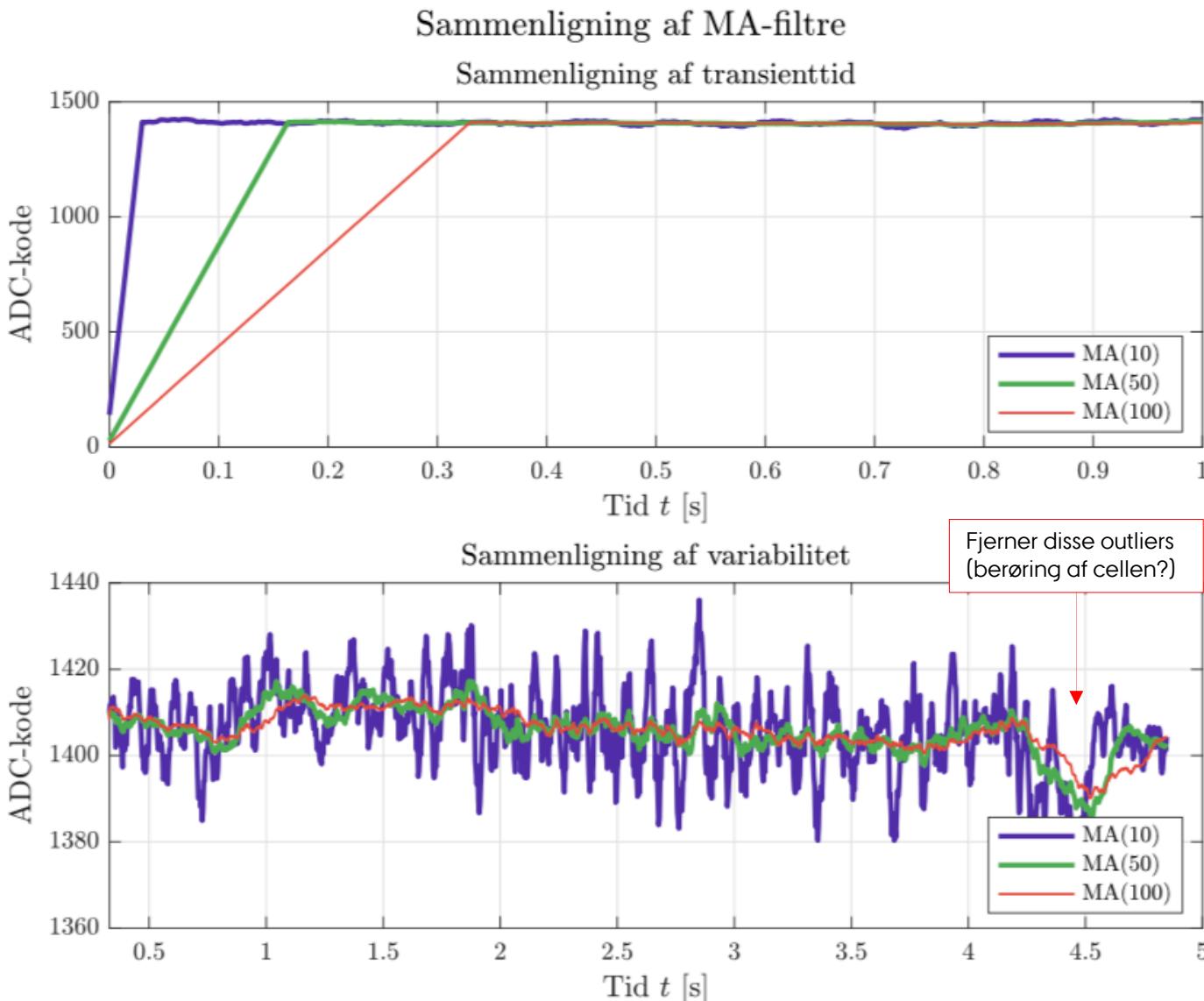
- Højere N:
 - Længere transientrespons
 - Bedre midling (lavere varians på output)

Fjerner nogle outliers og beregner reduktion i varians (effekt)

	Varians	Eff_reduk	Forv_reduk	Reduk_dB	Forv_dB
MA(10)	82.728	9.951	10	9.9787	10
MA(50)	16.372	50.282	50	17.014	16.99
MA(100)	11.36	72.466	100	18.601	20

Konklusion:

- Filtrene giver tilnærmet den teoretisk forventede reduktion i støjeffekt for MA(10) og MA(50)
- Ikke helt tilfældet for MA(100). Skyldes nok bl.a. outlieren ved omkring 2.8 s, der relativt set påvirker MA(100)-filteret i længere tid end de to andre filtre.



SAMMENLIGNING MED EKSPONENTIELT MIDLINGSFILTER

Dæmpning af støjeffekt [1, s. 610], og
støjreduktionsfaktor, R :

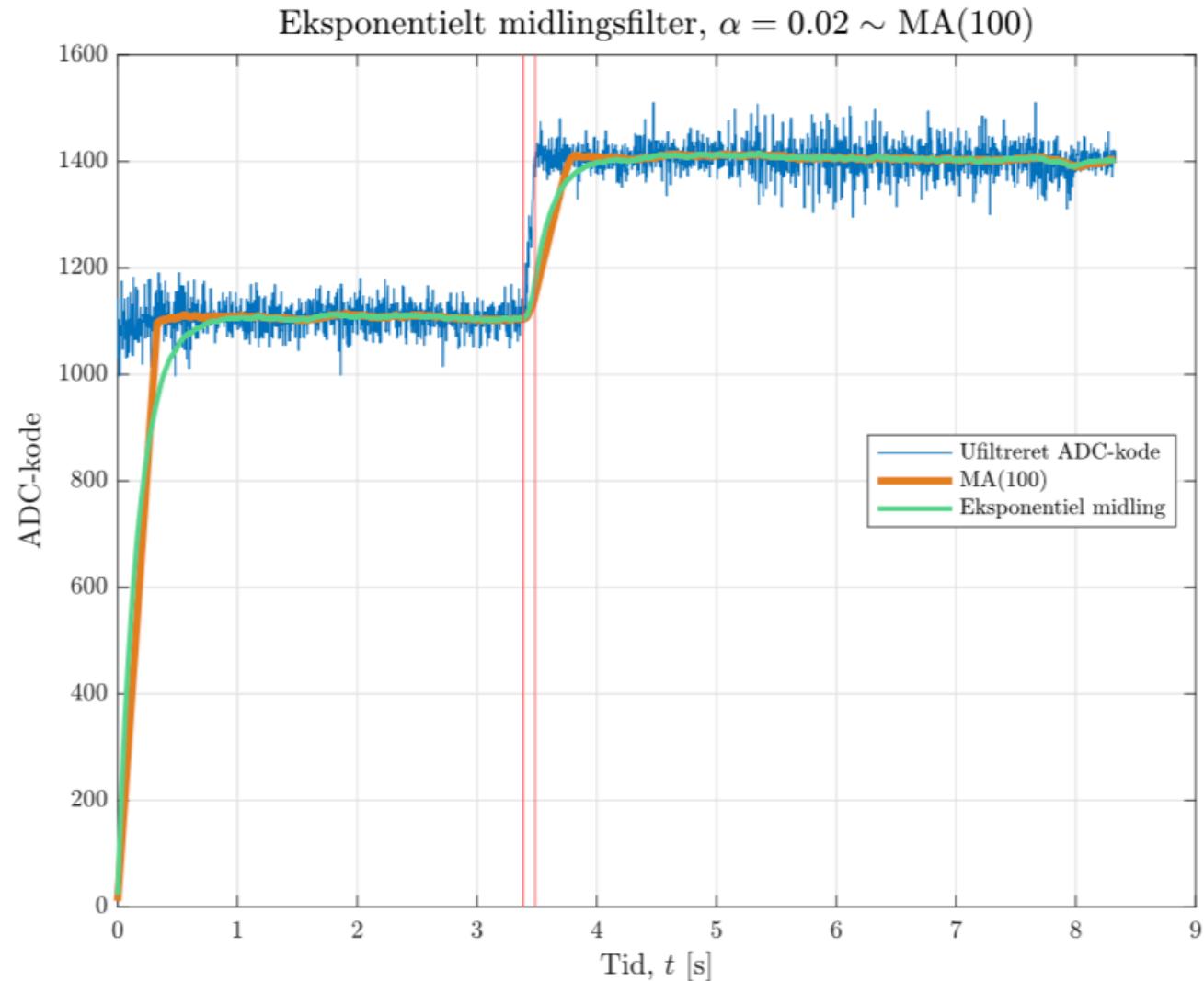
$$\frac{\hat{\sigma}_{\text{output}}^2}{\hat{\sigma}_{\text{input}}^2} = \frac{\alpha}{2 - \alpha} \quad R = \frac{2 - \alpha}{\alpha}$$

Eksempel: Et MA(100)-filter dæmper med faktor $R = 100$,
så $\alpha = 0,02$ kan emulere dette.

	Varians	Eff_reduk	Forv_reduk	Reduk_dB	Forv_dB
exp($\alpha=0.02$)	13.197	66.802	100	18.248	20
MA(100)	12.48	70.64	100	18.491	20
exp($\alpha=0.18$)	85.921	10.26	10	10.112	10
MA(10)	86.964	10.137	10	10.059	10

Resultater:

- Cirka samme støjdæmpning som tilsvarende MA-filter (som forventet)
- Langsommere indsvingning (lav α)
- "Uendelig" (ekspenentielt dæmpet) effekt af outliers



DET BEDSTE FRA BEGGE VERDENER?

Lad filterparameter være variabel!

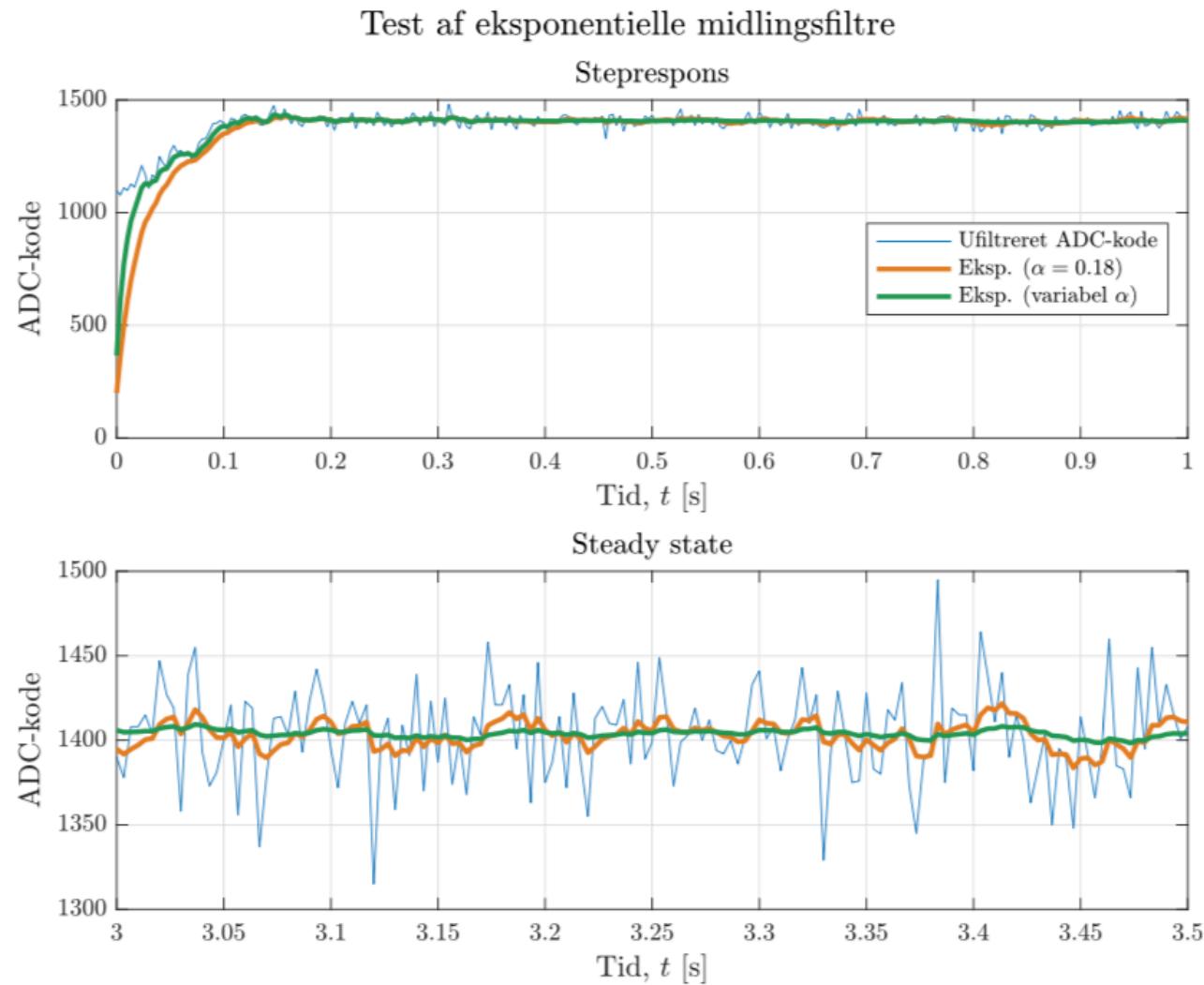
- Giver hurtig indsvingning og høj støjdæmpning

3 sektioner:

- 1: Samples 0-49, $R = 5 \sim MA(5)$
- 2: Samples 50-99, $R = 10$
- 3: Samples 100-, $R = 50 \sim MA(50)$
- $f_s = 300$ Hz i denne case.

Mulige hensyn ved design af algoritme til valg af sektioner og dæmpning:

- Krav til indsvingningstid
- Krav til støjdæmpning til forskellige tidspunkter



EKSEMPEL PÅ ANVENDELSE

User Experience-projekt (E4BIS)

Brugereksperimenter med fjernbetjent operationsudstyr

- Flex sensor -> servomotor
- Prototype med 50-taps rekursivt midlingsfilter -> Reaktion i realtid
- Forskellige slags feedback:
 - Visuel
 - Taktil / kraft
 - Haptisk



OPSUMMERING OG PERSPEKTIVERING

Overblik over statistiske mål for støj

Designovervejelser ved design af midlingsfiltre

- MA-filtre versus eksponentielle
- Mulighed for design af fleksibel algoritme, "det bedste fra begge verdener".

Øvrige muligheder

- Tids-synkron midling (parallel)
- PSD

6. MULTIRATE

1. Hovedpointer
2. Decimering
3. Interpolation
4. Resampling
5. Opsummering

Spørgsmål 6

Diskutér decimering og interpolation. Herunder nedsampling, opampling, spektre og filtre.

HOVEDPOINTER

Decimering og interpolation -> multirate-systemer

Multirate-systemer: To eller flere forskellige samplingsfrekvenser i et system [1, s. 511]

Formål:

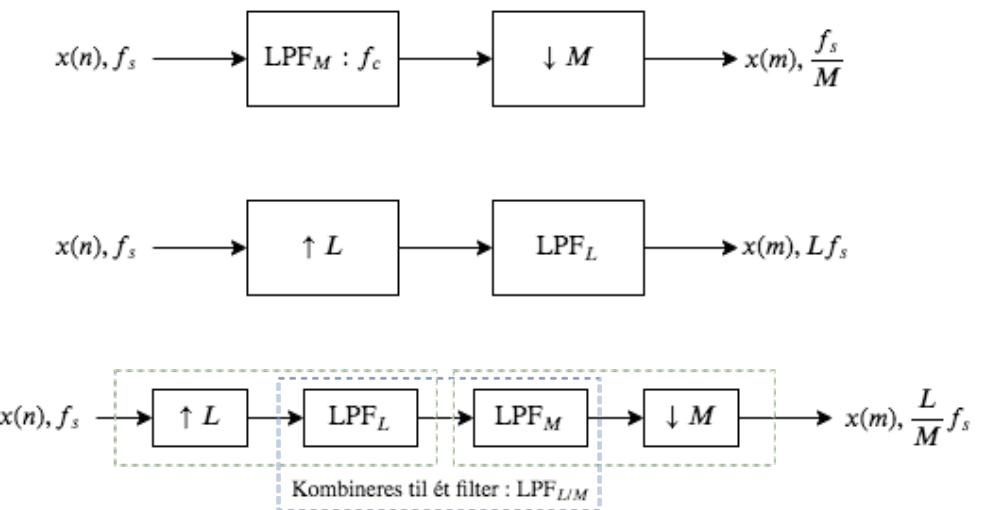
- Ændre samplingsfrekvensen for et signal ned eller op
- Ny sampling af signal ikke mulig

Hvorfor:

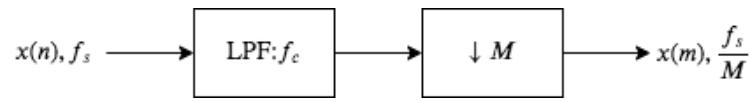
- Multi-rate-systemer, fx realtid med to forskellige processorer, el. D/A med anden frekvens end A/D, DSP, el. lign.
- Reducere beregningskompleksitet
- Reducere antal filterberegninger for et bånd-begrænset signal

Terminologi:

- Decimering: Sænke samplingsfrekvens, faktor M .
- Interpolation: Øge samplingsfrekvens, faktor L .
 - I den proces beregnes/interpoleres mellemliggende samples.
- Resampling: Ændre samplingsfrekvens med rational faktor $\frac{L}{M}$.



DECIMERING



Decimering \equiv lavpasfiltrering og downsampling

Nedsampler (downsampler) med faktor M

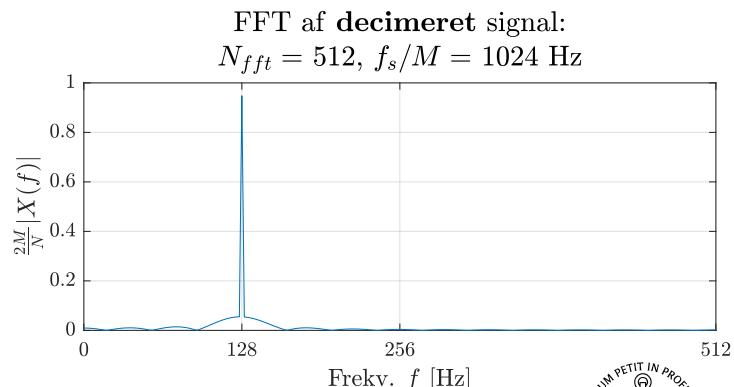
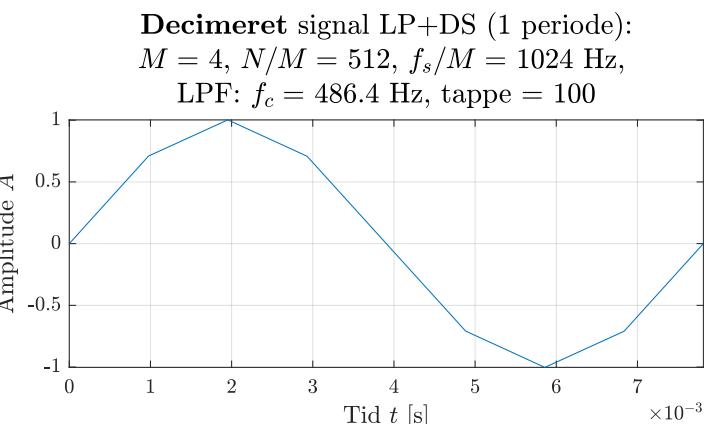
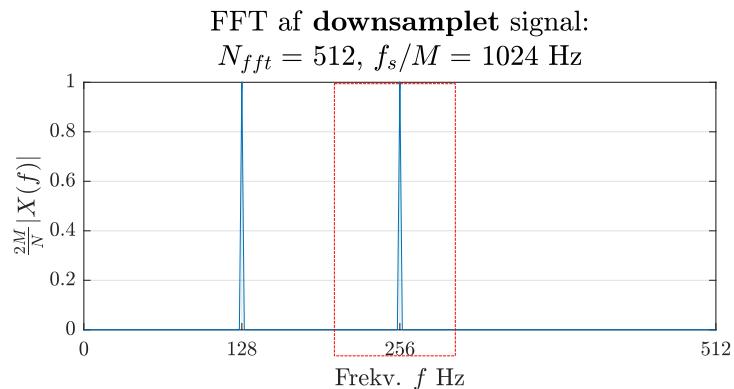
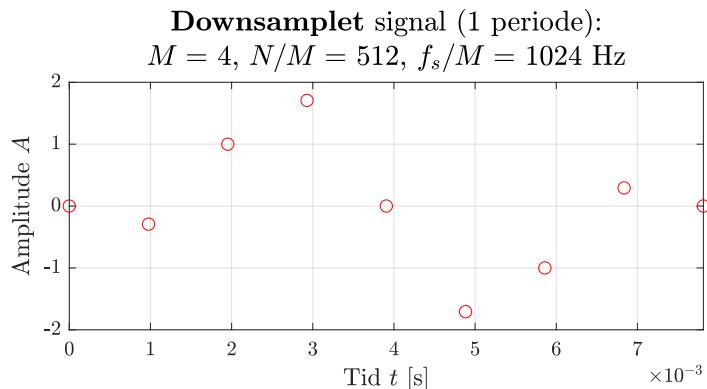
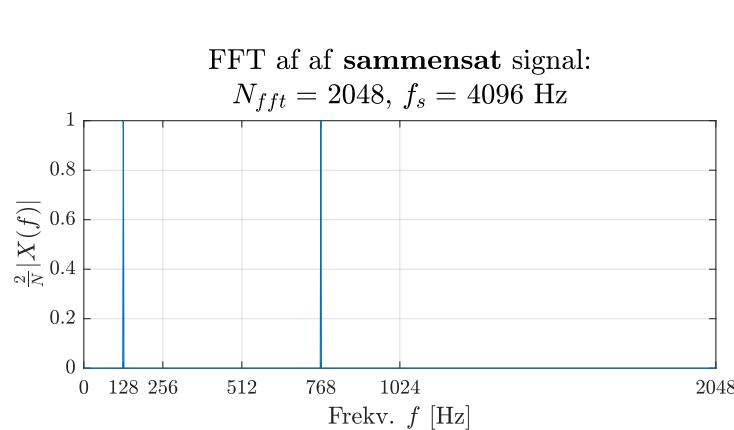
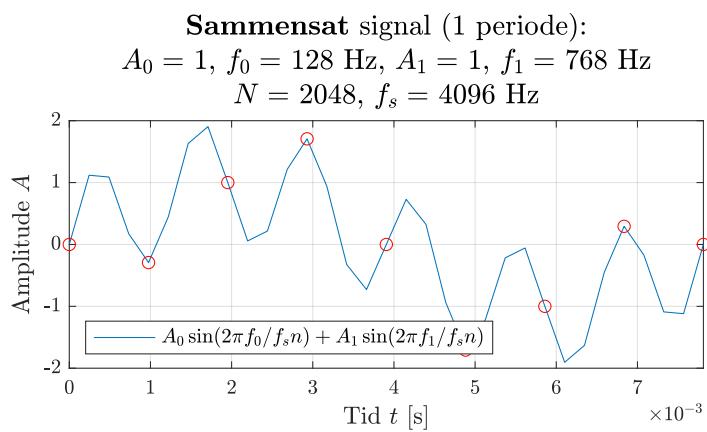
- Beholder hver M'te sample
- Ikke tidsinvariant operation!
- Faktor M-dæmpning i frekv.dom. [1, s. 515]
- Ingen dæmpning i tidsdomæne

Shannon/Nyquist skal gælde både før og efter

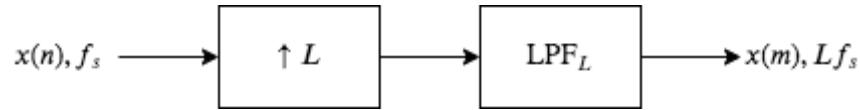
- $f_s \geq 2B$

Lavpasfiltrering:

- Behov for LPF, ellers spektral aliasering
 - FIR: Lineær fase
 - $f_c = \frac{f_{s,ny}}{2} = \frac{f_{s,gl}}{2M}$ (ideelt)
 - I praksis vælges f_c mindre,
 - Fx $f_c = \alpha \frac{f_{s,gl}}{2M} = 95\% \cdot \frac{4096}{2 \cdot 4} = 486$ Hz.



INTERPOLATION



Interpolering \equiv opsampling/zero-stuffing og LP-filtrering:

Zero-stuffing

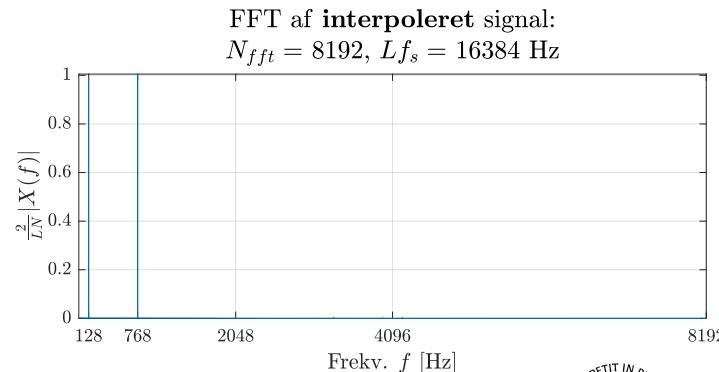
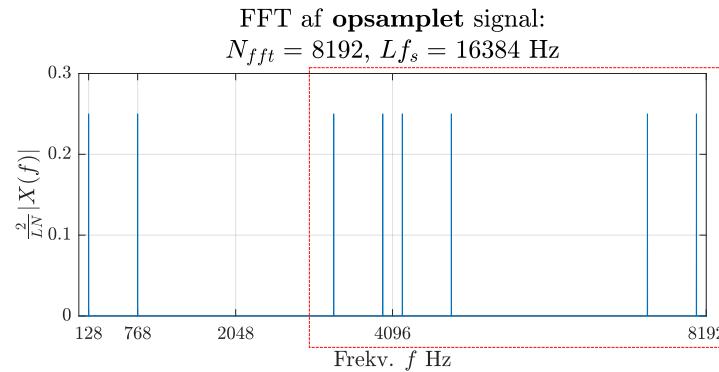
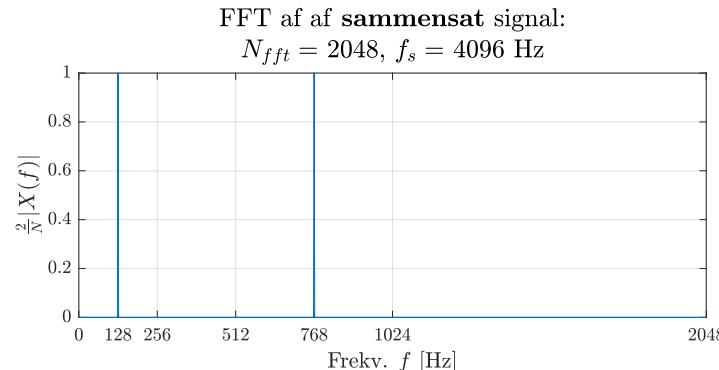
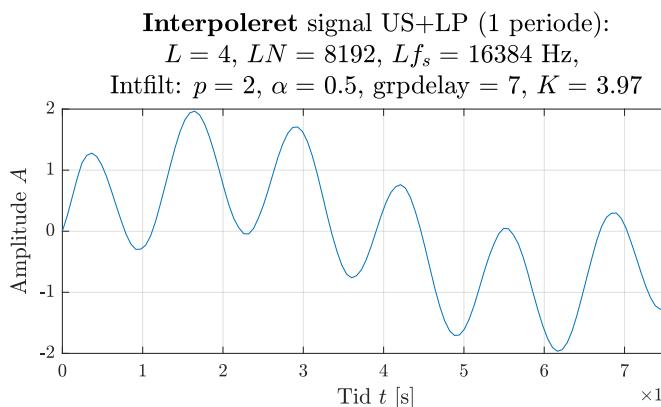
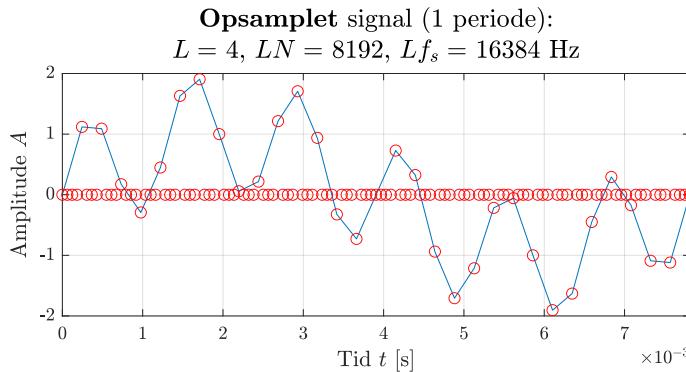
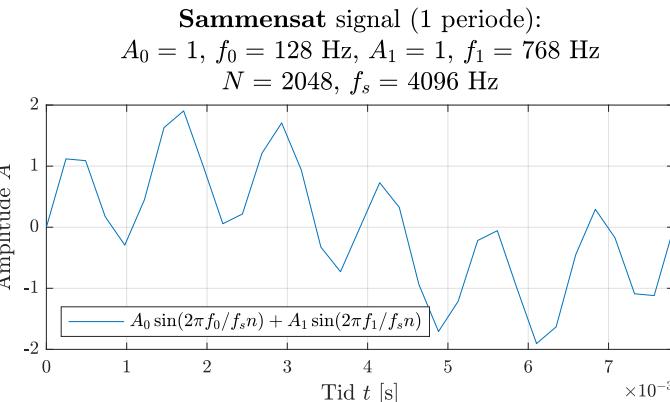
- Zero-stuffing med faktor L , indsætter $L - 1$ nul-samples for hver oprindelig
- Introducerer nye spektrale images
- Processen giver faktor L amplitudetab [1, s 519]
- Skal opvejes af gain, K , i interpolationsfilter.

Lavpasfiltrering:

- Beregne værdier for $L - 1$ nye mellemliggende sampleværdier (interpolerede værdier \rightarrow midlingsfilter)
- LPF dæmper spektrale images
- LPF er en midlingsberegning.

LPF'et er interpolationsfilter:

- Filter er ikke-ideelt, stadig residuale spektrale images.
- Kvalitet af interpolering afhænger af kvalitet af interpoleringsfilteret.



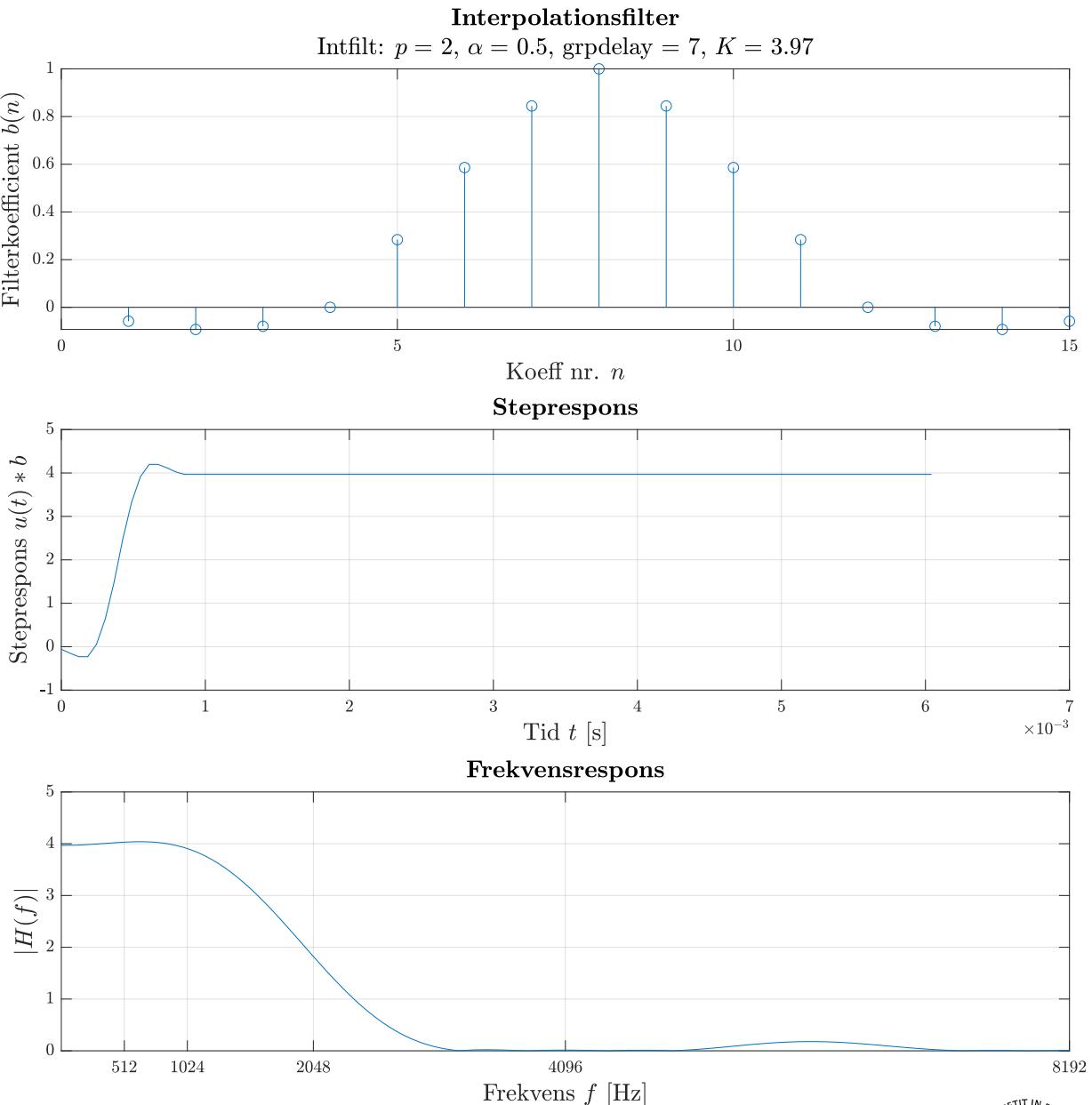
INTERPOLATIONS-FILTER

Interpolationsfilter

- FIR-lavpasfilter med gain.

Implementering:

- Designes med $\text{intfilt}(L=4, p=2, \alpha=0.5)$
- Gain: ~ 4
 - $K = \frac{\sum b_i}{\sum a_i}$ [1, s. 300]
- Benytter nærmeste $2p$ ikke-nul samples til interpolation (kunne også have brugt interp)
- Bandlimited inputsignal, oprindeligt $x(n)$ optager under halvdelen af Nyquist-intervallet, så
 - $\alpha = 0.5$, Bandlimited til: $\alpha \frac{f_{s,gl}}{2} = \frac{1}{4} f_{s,gl} = 1024$ Hz.
- Længde: 15
- Group delay: 7



RESAMPLING

Opnå en ny samplingsfrekvens

- Rationel brøk: $f_{s,ny} = \frac{L}{M} f_{s,gl}$
- Sammensæt interpolering og decimering.

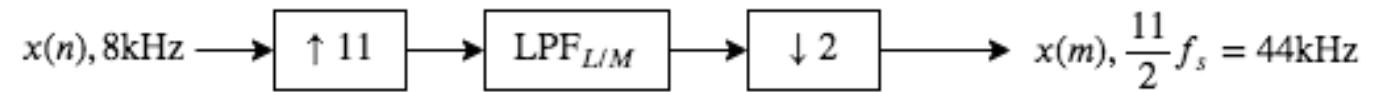
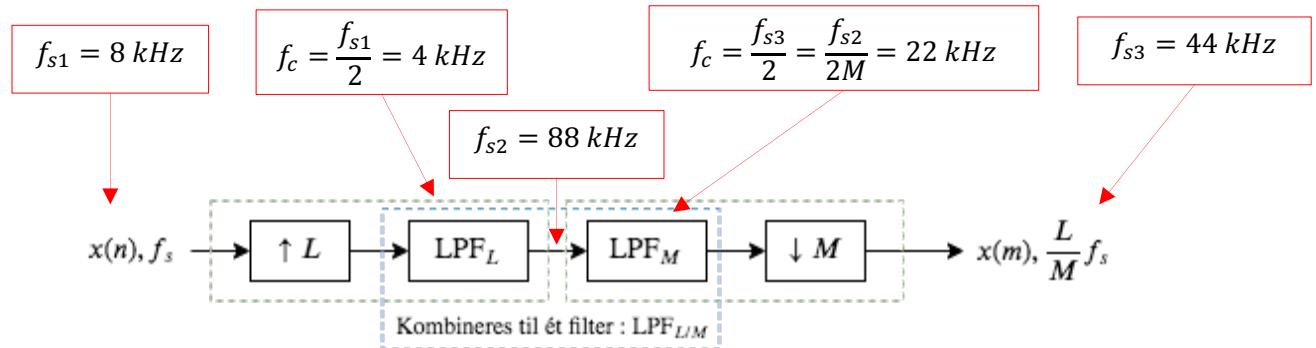
Designmål for multirate-system

- SW/HW betragtninger
 - Antal beregninger ->
 - Hurtige filter (lav orden)
 - Multiplikationer/sek. (mac-operationer)
 - Billigere DSP, mindre strøm
 - Kvalitet: Minimal aliasering (spektrale images)

Generelt er denne slags multi-rate beregningsmæssigt ineffektivt

- LP-filtrerer en masse samples og så smides de væk bagefter i nedsamlingen
- Mere effektive filtre:
 - Reducér antal tappe i FIR-filter. Metoder:
 - Two-stage: $M = M_1 M_2$
 - Polyphase

Eksempel, opg. 12: 8 kHz signal til 44 kHz D/A



Kan "nøjes" med et enkelt multirate-filter med $f_c = \frac{f_{s1}}{2} = 4 \text{ kHz}$

OPSUMMERING

Motivation til at undersøge multi-rate-systemer

Metode til decimering med faktor M, vigtigt at undgå aliasering

Metode til interpolation med faktor L, vigtigt at undgå spektrale images

Lidt om interpolationsfilteret

Sammensætning af et resamplingsystem med faktor L/M.

7. FOLDNING OG KORRELATION

1. Overblik: Foldning og korrelation
2. Foldning og foldningssætningen
 1. Eksempel med vinduer i frekvensdomæne
3. Korrelation
 1. Eksempler fra Case 4
4. Opsummering

Spørgsmål 7

Diskutér foldning og korrelation. Referér eventuelt til Case-projekt 4.

OVERBLIK: FOLDNING, KORRELATION

Begreb

Formel

Foldning

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

Hvis kausale, så kun fra $k = 0$.

Kryds-korrelation

$$r(n) = \sum_{k=-\infty}^{\infty} h(k)x(n+k) = h(n) * x(-n)$$

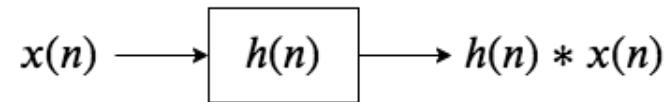
Auto-korrelation

$$r_x(n) = \sum_{k=-\infty}^{\infty} x(k)x(n+k) = x(n) * x(-n)$$

Korrelation med n -lagget version af sig selv

Anvendelse

Digitalt LTI-system: Filtrering, hvor $x(n)$ er input og $h(n)$ er impulsrespons



Sammenligne to signaler
Finde ét signal i et andet signal

Sammenligne et signal med sig selv
Periodicitet
Relaterer til systemidentifikation
• AR, MA, eller ARMA-modeller

FOLDNINGS-SÆTNINGEN

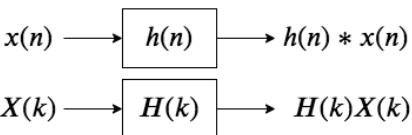
Hovedresultat

For LTI-systemer; DTFT, DFT (og FFT), z-transformation, mv.

Foldning i tidsdom. \leftrightarrow multiplikation i frekvensdom.

$$h(n) * x(n) \xrightarrow{DFT} H(k) \cdot X(k)$$

Anvendelse: Effektiv filtrering vha. DFT/FFT

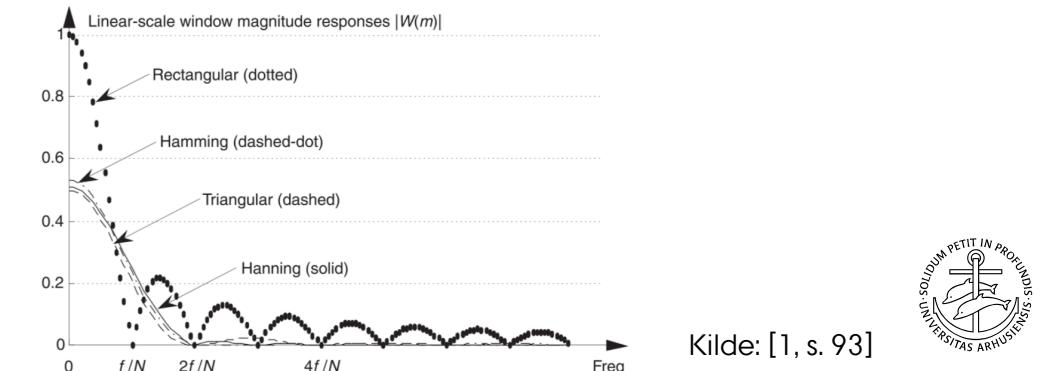
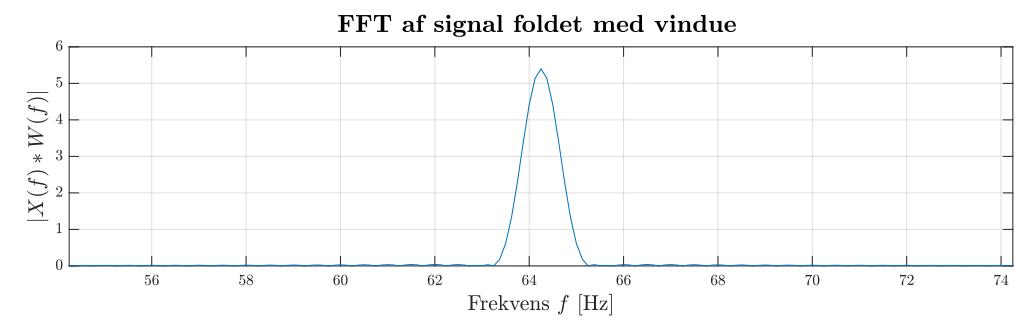
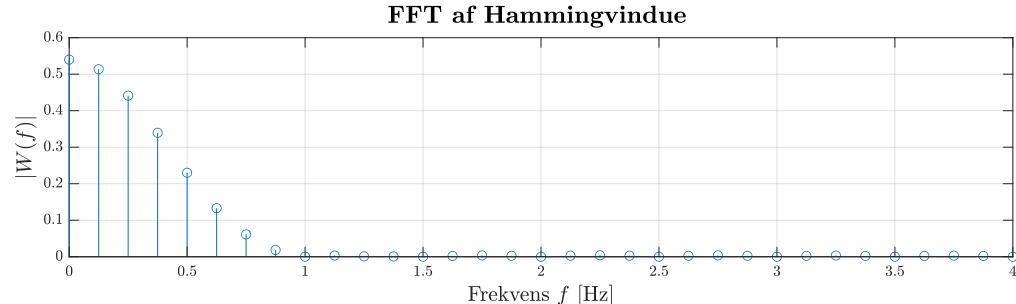
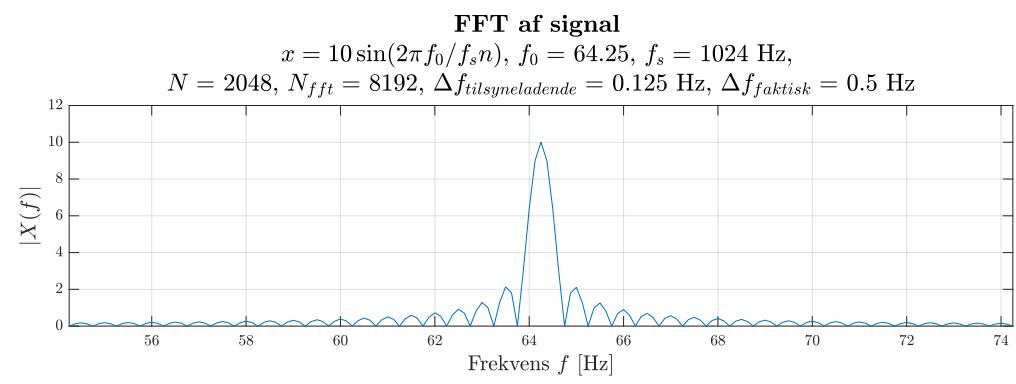


Multiplikation i tidsdom. \leftrightarrow foldning i frekvensdom.

$$h(n) \cdot x(n) \xrightarrow{DFT} H(k) * X(k)$$

Anvendelser:

- Vinduer \rightarrow foldning med smallt filter i frekvensdom.
- Modulation, fx AM. "Frekvensshifting".



CASE 4: ANVENDELSE AF KORRELATION

Kort fortalt...

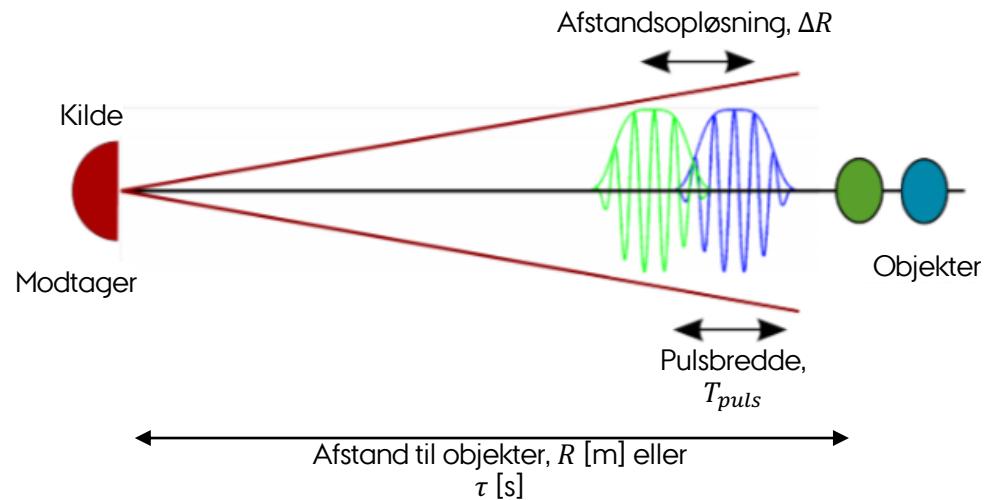
Sonar-system

- Puls fra kilde -> refleksion -> modtager
- Afstand i tid og meter findes ved krydkorrelation mellem afsendt puls og modtaget signal.
- Samplede signaler, tidsforskydning $\tau = n_\tau T_s$

$$R = \frac{v \cdot n_\tau \cdot T_s}{2} = \frac{v \cdot n_\tau}{2f_s}$$

- Lydbølge i luft, $v = 343$ [m/s] ved 20°C

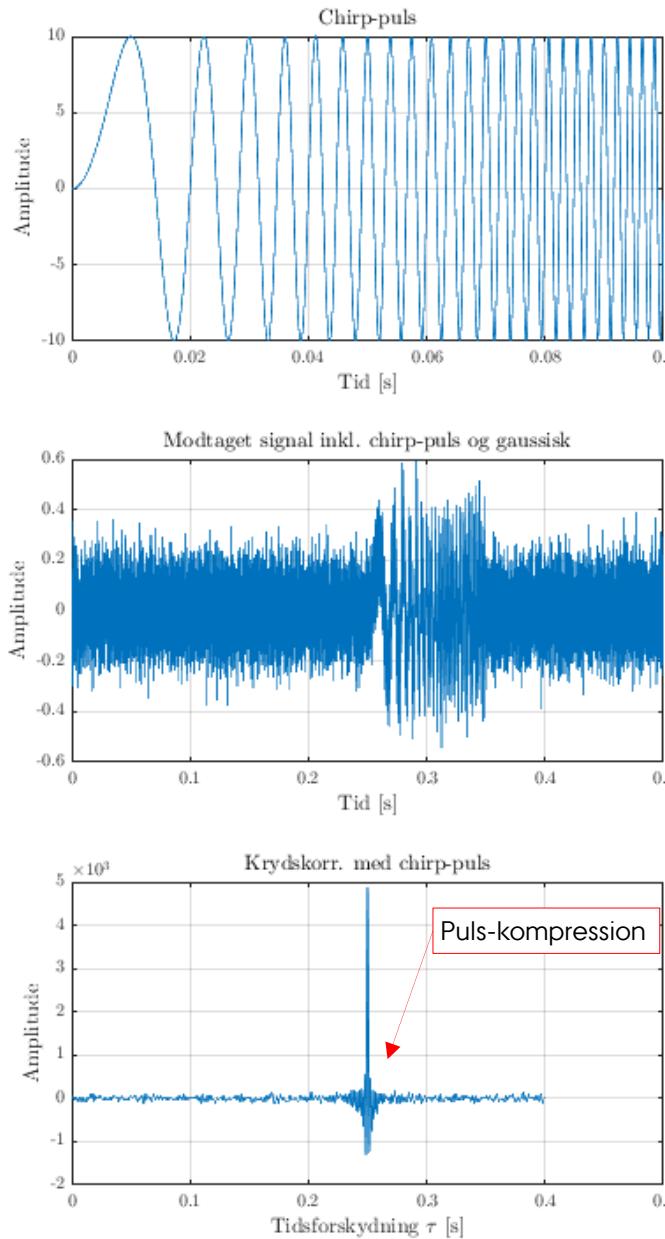
Trade-off:
Brug for "meget" data til at lave god korrelation - men puls så kort som muligt for at få god afstandsopløsning:
Øg pulsfrekvens - men LPF i luft?



Problemet er

- Bestemme n_τ vha. krydkorrelation
- Finde pulstype, der giver præcise resultater

Simulering af sonar
 $A = 10$, $R = 43$ [m], $\sigma_r = 0.1$

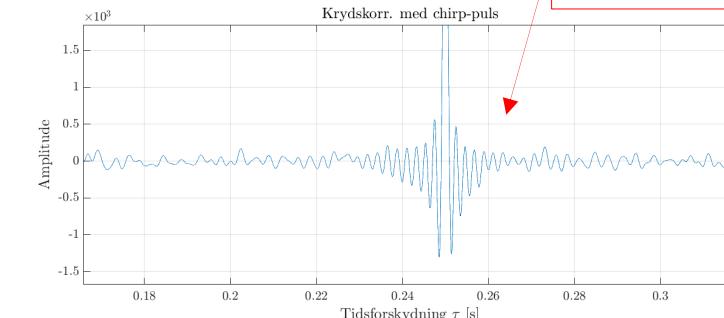


Chirp FM 0-500 Hz. Sinus ren 500 Hz.

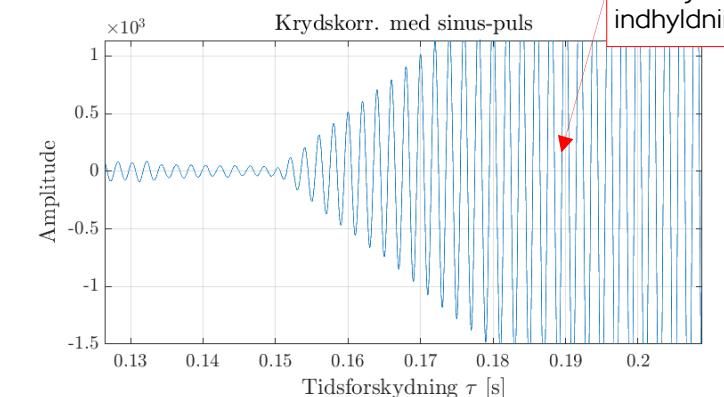
Konklusion:

- Frekvensmoduleret puls fungerer bedst.

Chirp korreleret ned chirp giver sinc-agtig



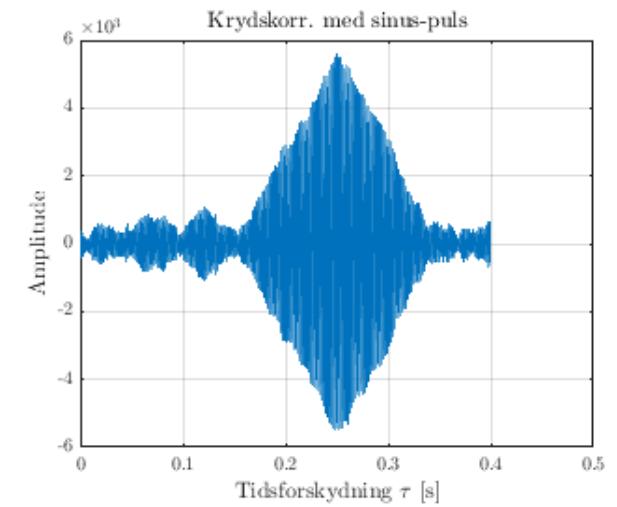
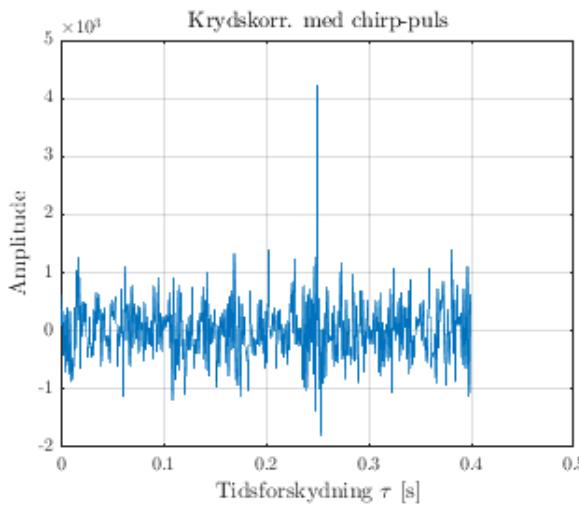
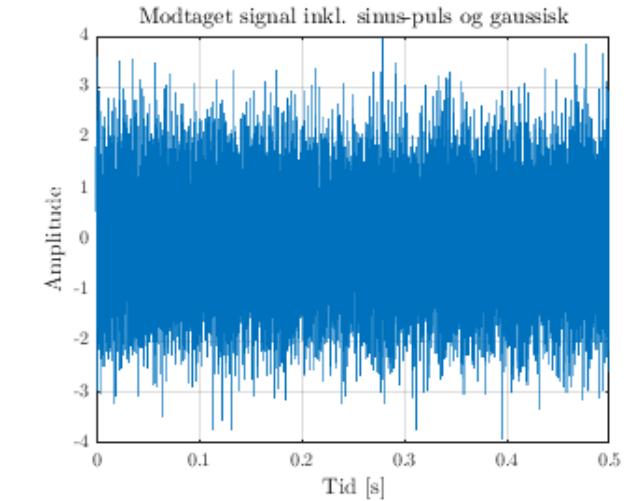
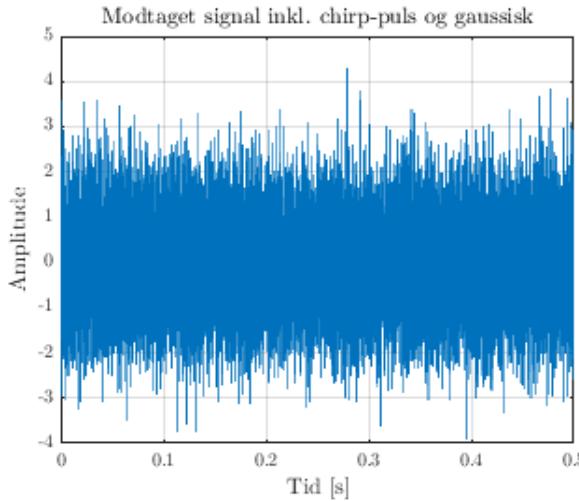
Sinus korreleret med sinus giver sinus med indhyldning



FAKTOR 10 MERE STØJ

- I tidsdomænet kan pulsen ikke længere direkte ses for støj.
- Med pulskompression kan afstanden bedømmes præcist selv med væsentlig støj.
- For sinus-pulsen er billedet noget mere “mudret”

Simulering af sonar
 $A = 10$, $R = 43$ [m], $\sigma_r = 1$



LUFTENS LPF

Undersøge tradeoff ift. at luften lavpasfiltrerer

Højere frekvens i pulserne:

- Generelt bedre performance for “sonar”-systemet.
- Hvor højt man kan presse puls-frekvensen for de to forskellige sonar-typer givet usikkerhed på luftens knækfrekvens?

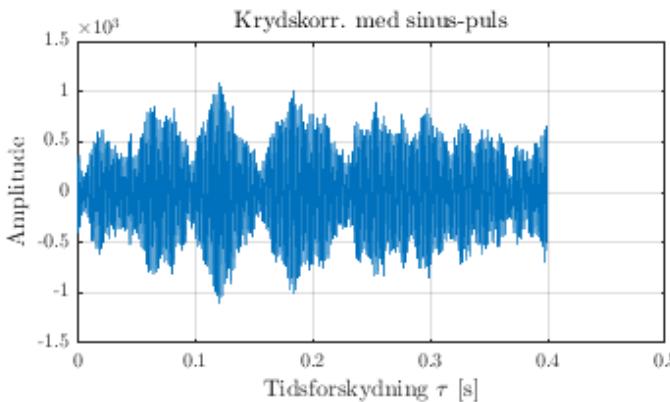
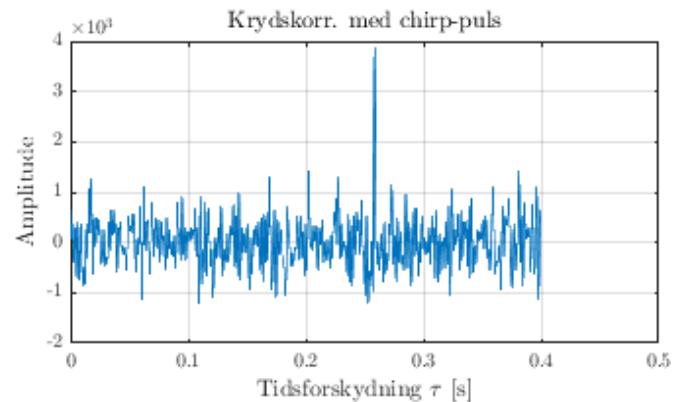
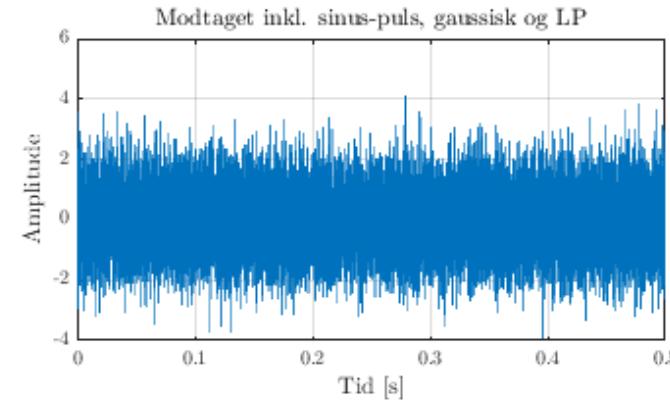
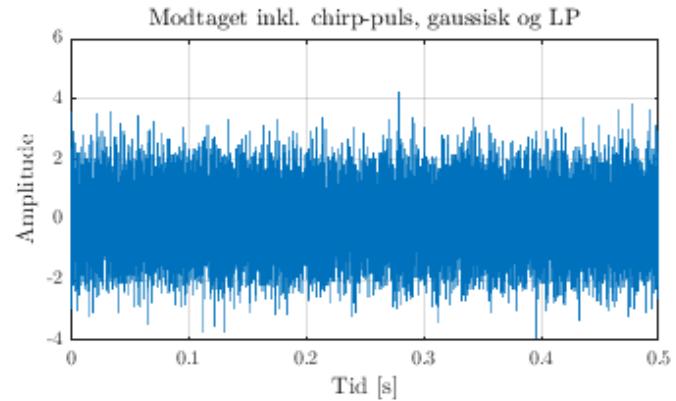
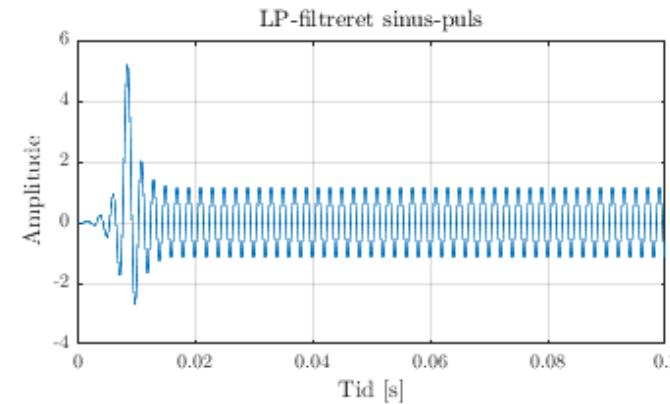
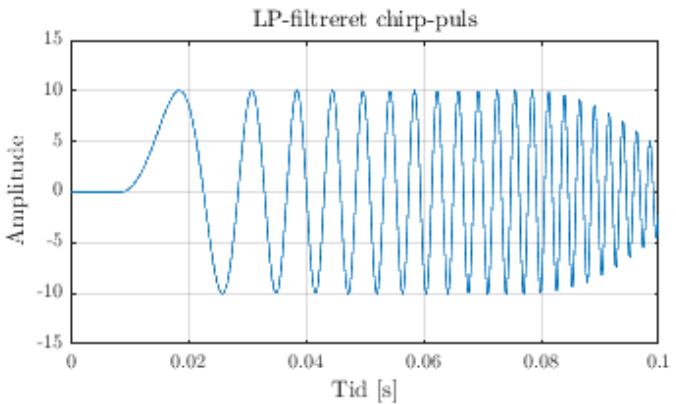
Antag, at knækfrekvensen i luften ligger omkring F1 (500 Hz) men med 10 pct. usikkerhed.

- LP-filteret får en worst-case knækfrevens på 0.9F1.

Konklusion

- LP-filtrering værst ud over sinus-pulsen, som ikke længere kan give et præcist estimat på targets afstand.
- “falsk-positiv” ved $\tau = 0.12$ s, dvs. $R = 21$ m

Simulering af sonar
 $A = 10$, $R = 43$ [m], $f_c = 450$ [Hz], $\sigma_r = 1$



OPSUMMERING

Overblik over foldning og korrelation – og hvordan de er relateret

Eksempel på anvendelse af foldning og foldningssætningen

- Vinduer i frekvensdomænet

Eksempel på anvendelse af krydskorrelation

- Sonar-system,
 - FM-chirp giver puls-kompression
 - Bedste støjimmunitet, kan "presse" systemet "hårdest" med FM chirp.

8. DIGITAL DIFFERENTIATION OG INTEGRATION

Spørgsmål 8

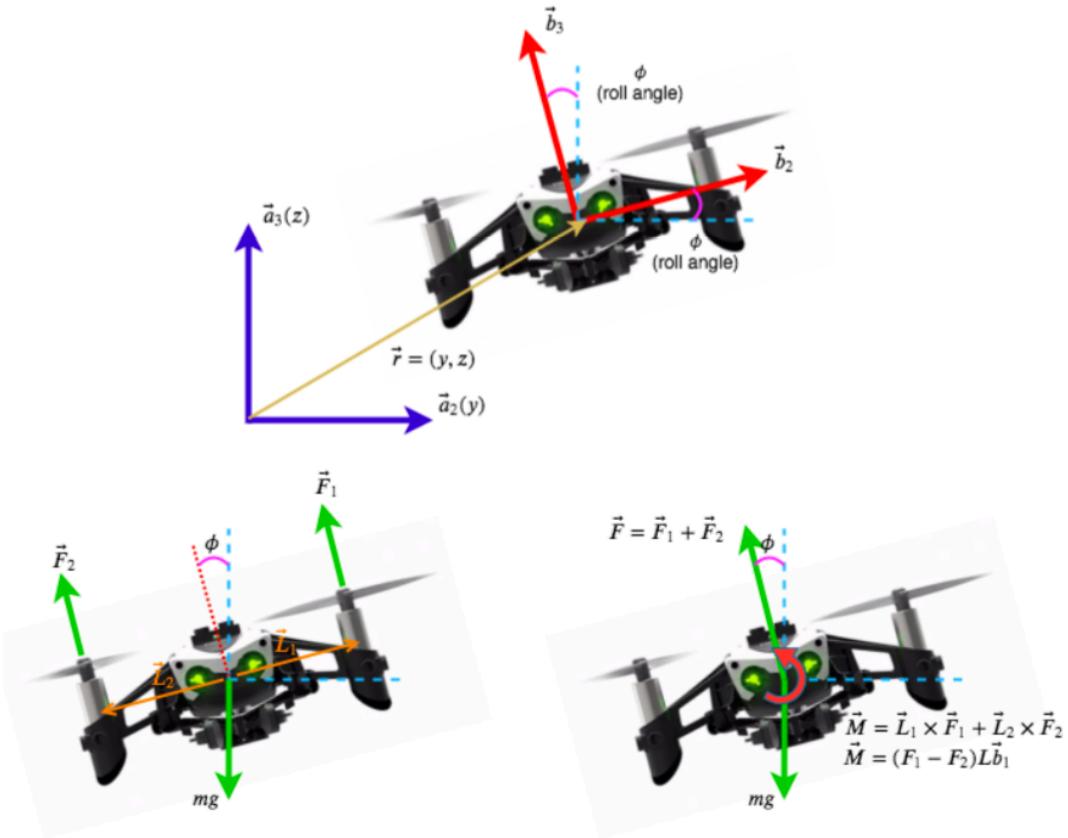
Diskutér digital differentiation og integration. Sammenlign forskellige metoder.

1. Motivation
2. Digital differentiation
 1. Overblik
 2. Sammenligning af forskellige typer
3. Digital integration
 1. Overblik
 2. Sammenligning af forskellige typer
4. Opsummering

MOTIVATION

Reguleringssteknik / digital kontrol

- Implementering af dynamiske systemer / bevægelsesligninger
 - State-space repræsentationer
- Input fra sensorer
 - Accelerationer fra IMU integreres op til hastighed, position, Euler-vinkler og vinkelhastigheder i forskellige referencerammer
 - Støjimmunitet er vigtigt
- Regulatorer
 - PD-regulator benytter *derivative gain*, dvs. førsteordens afledeede på fx en positionsfejl benyttes som input til en controller
 - PID også med *integral gain*.



$$\begin{bmatrix} \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} -\frac{1}{m} \sin(\phi) F \\ -g + \frac{1}{m} \cos(\phi) F \\ \frac{1}{I_{xx}} M \end{bmatrix} = \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix} + \begin{bmatrix} -\frac{1}{m} \sin(\phi) & 0 \\ \frac{1}{m} \cos(\phi) & 0 \\ 0 & \frac{1}{I_{xx}} \end{bmatrix} \begin{bmatrix} F \\ M \end{bmatrix}$$

$$h(t) = K_p e(t) + K_d \dot{e}(t) = K_p(z_{\text{ref}} - z(t)) + K_d \frac{d}{dt}(z_{\text{ref}} - z(t))$$

DIGITAL DIFFERENTIATION

Differentiation er veldefineret i kontinuert tid / analogt

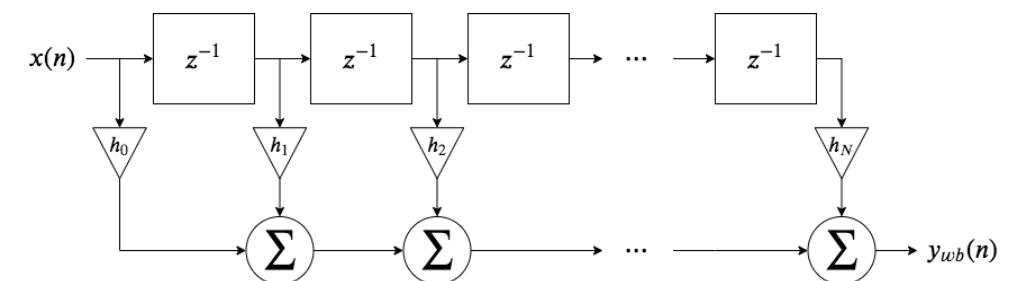
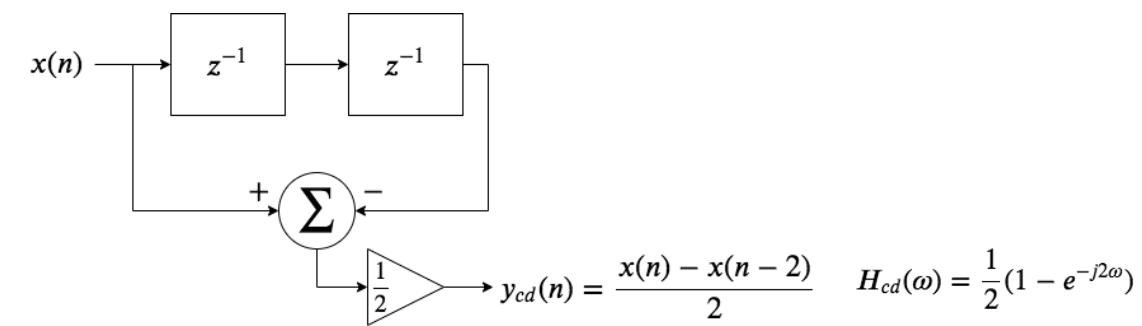
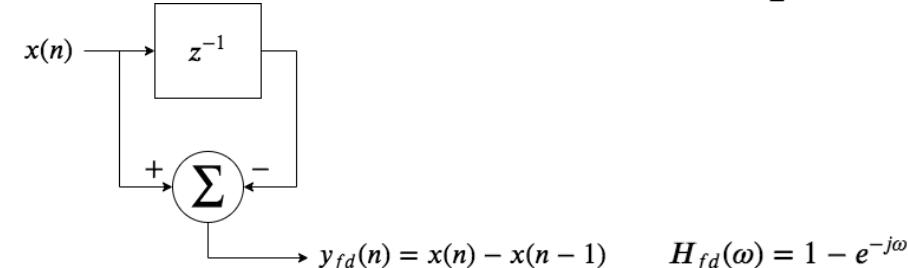
Forskellige måder at definere differentiation i diskret tid / digitalt.

- Her FIR-filter-typer
 - First-difference ('digital differencer')
 - Central-difference (kausal)
 - Wideband (koefficienter skal designes)

Forskellige karakteristika ift.

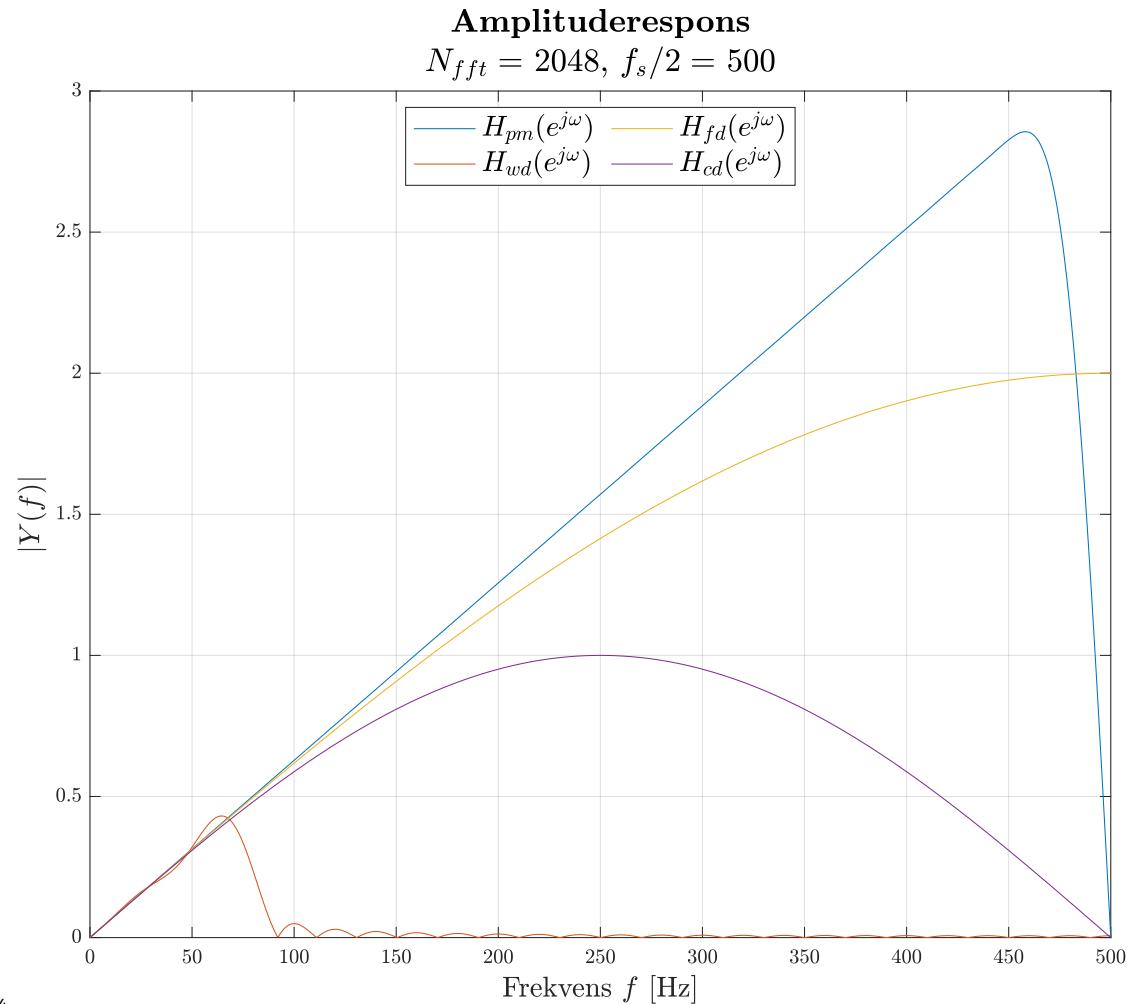
- Effektivt frekvensområde for input
- Frekvensrespons
- Støjimmunitet, støjdæmpning

$$e^{j\omega t} \rightarrow \boxed{\text{Ideel differentiation: } H(\omega)} \rightarrow H(\omega)e^{j\omega t} = j\omega e^{j\omega t}$$
$$H(\omega) = j\omega$$
$$|H(\omega)| = |\omega|$$
$$\angle H(\omega) = \text{sgn}(\omega)\frac{\pi}{2}$$



OVERBLIK: FILTERTYPER TIL DIGITAL DIFFERENTIERING

Type	Fordele	Ulemper
First difference	Simpel at implementere	Kan øge støj på outputtet <ul style="list-style-type: none">Forstørker højfrekvent støj
Central difference	Dæmper højfrekvent støj	Kræver et memory-element mere
Wideband	Knækfrekvens er designparameter <ul style="list-style-type: none">Kan tune specifikt frekvensbånd eller til at være tæt på ideel Orden er designparameter	Længere group delay ved højere filterorden Mere kompliceret at implementere <ul style="list-style-type: none">Flere memory-elementer Kan øge støj på outputtet



CASE: FD OG CD

Fysisk model: Finde hastighed, $v(t)$, fra målt position, $x(t)$

- $x(t) = \int_0^t v(\tau)d\tau, v(t) = \dot{x}(t) = \frac{dx(t)}{dt}$.
- Hvis $v(t) = k \rightarrow x(t) = kt$ (rampe):
- Samler $x(t)$ på $t = nT_s$ med $f_s = 1/T_s$, så $x(n) = knT_s$
 - First diff.: $y_{fd}(n) = x(n) - x(n-1) = kT_s$
 - Central diff.: $y_{cd}(n) = \frac{1}{2}(x(n) - x(n-2)) = kT_s$
- Derfor beregnes hastigheden vha. $v = y_{fd}(n)f_s = y_{cd}(n)f_s$.
- Analyse med additiv støj (wideband-støj, gaussisk hvidstøj)

Filterdesign:

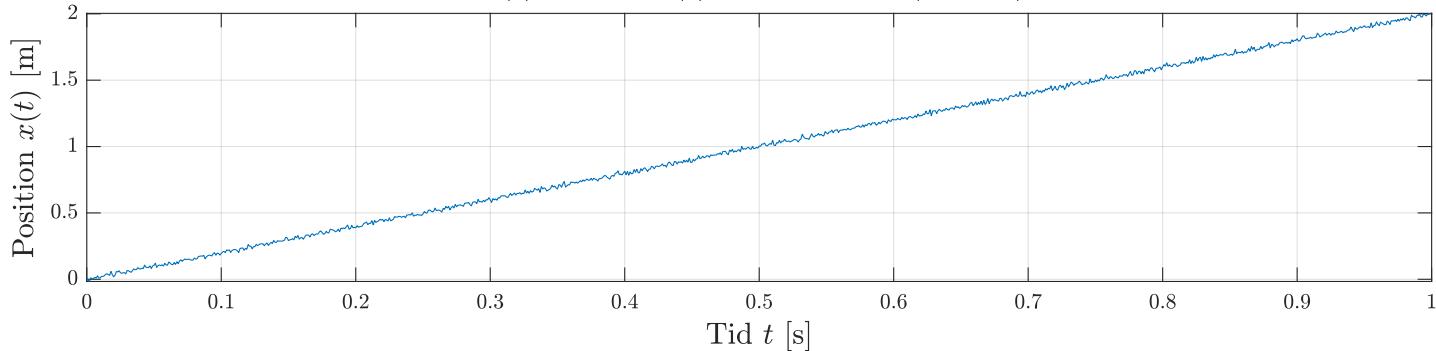
- Koefficienter aflæses direkte fra flow-diagram el. ovf.fkt.
 - First diff.: $b_{fd}(n) = [1, -1]$
 - Central diff.: $b_{cd}(n) = \frac{1}{2}[1, 0, -1]$

Konklusioner:

- Estimater på hastighed næsten ens
- Støjimmunitet:
 - Begge typer dæmper LF-støj
 - FD forstærker højfrekvent støj
 - CD dæmper højfrekvent støj

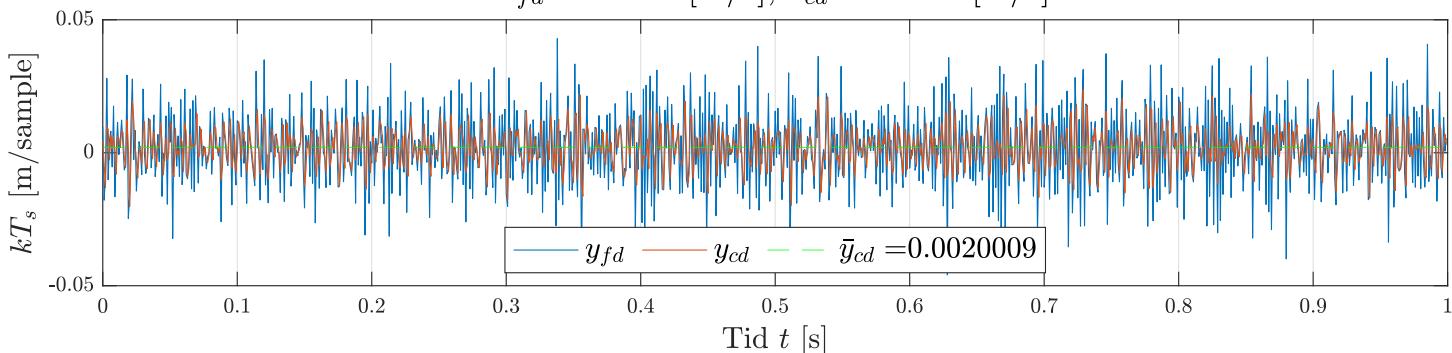
Samplet position

$$N = 1000, f_s = 1000 \text{ Hz}, \\ x(t) = kt + \epsilon(t), k=2, \epsilon \sim N(0, 0.01)$$



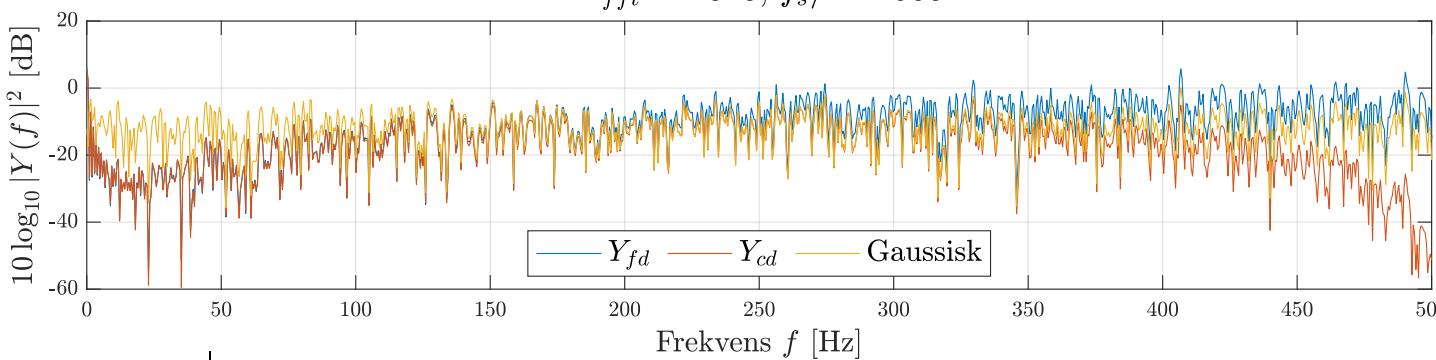
Filtreret signal og hastighedsestimater

$$\hat{v}_{fd} = \hat{k} = 2 \text{ [m/s]}, \hat{v}_{cd} = \hat{k} = 2 \text{ [m/s]}$$



Effektspektrum

$$N_{fft} = 2048, f_s/2 = 500$$

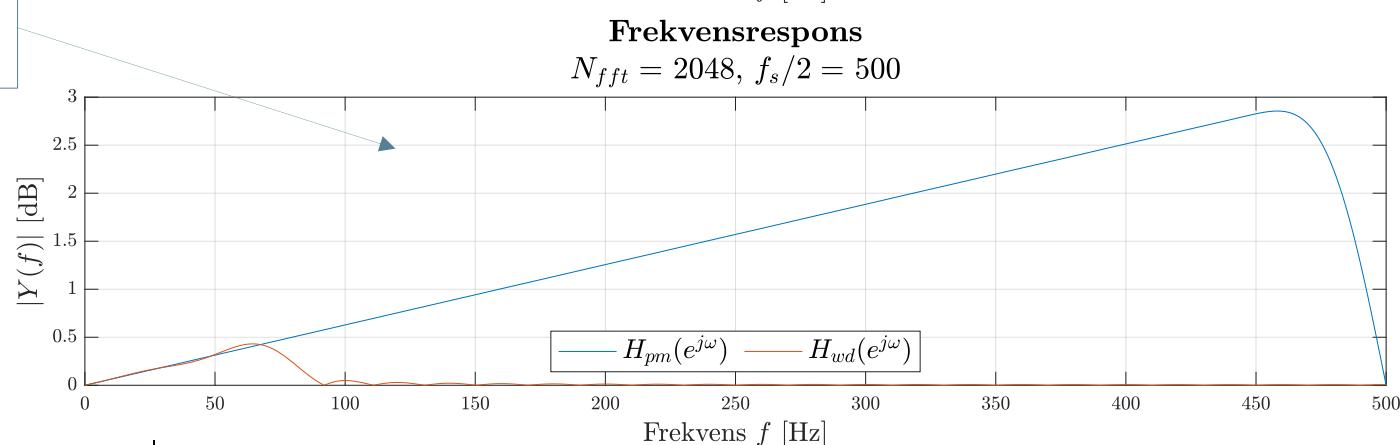
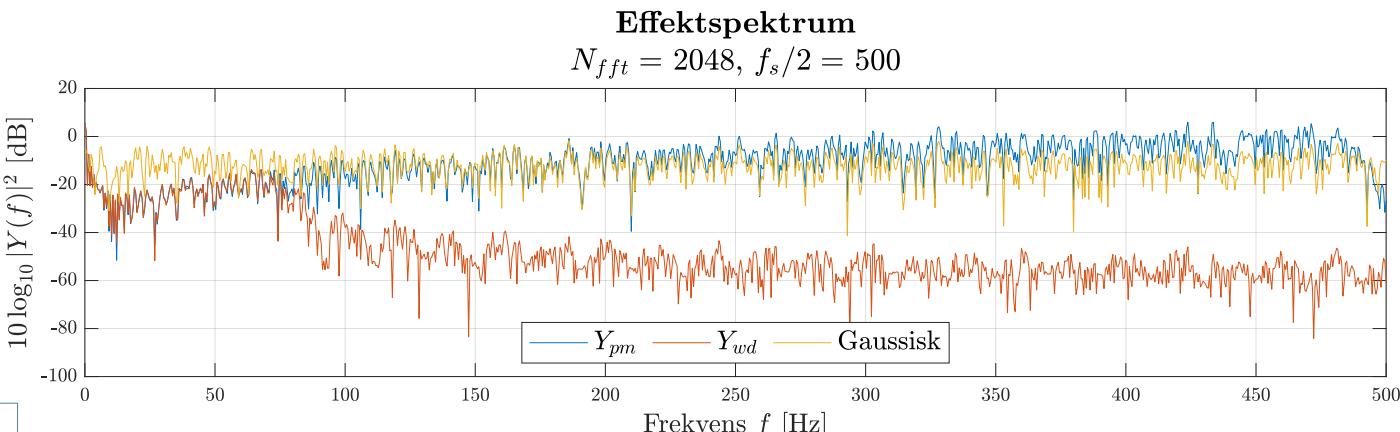
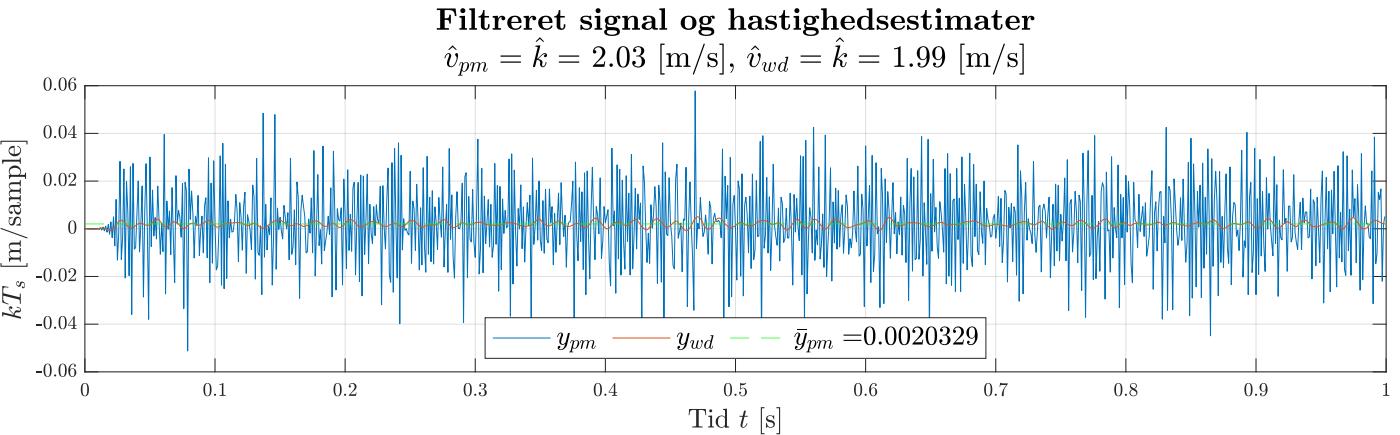


CASE: WIDEBAND

Samme fysiske problem som før:

- Design af wideband differentiationsfilter, forskellige metoder:
- **Lyons 7-15:** $N < 25$, ω_c som designparameter [1, s. 369]
 - $h(k) = \frac{\omega_c \cos(\omega_c[k-M])}{\pi[k-M]} - \frac{\sin(\omega_c[k-M])}{\pi[k-M]^2}$
 - Lad $0 \leq k \leq N - 1$
 - Lige N , "Center"-koeff. $M = (N - 1)/2$
 - Ulige N , centerkoeff. sættes: $h\left(\frac{N-1}{2}\right) = 0$
- **Parks-McClellan-algoritme:** $N > 25$:
 - Fx med *firpm* og type 'differentiator'
 - *firpm(50, [0 0.9], [0 0.9*pi], 'differentiator')*
- Lyons 7-14 (ej vist): $\omega_c = \pi$ ($f_c = \frac{f_s}{2}$), N ulige.
 - $h(k) = \frac{(-1)^k}{k}$

Pm: $\omega_c = 0.9\pi$ (90% af f_s)
 Wd: $\omega_c = 0.5$ (16% af f_s)
 Orden: 50 for begge



Konklusioner:

- Cirka samme estimater som før
- Kan komme tættere på en ideel diff. eller designe specifik båndbredde
 - ω_c er designparameter
- Konsekvenser for støjimmunitet.

DIGITAL INTEGRATION

Integration er veldefineret i kontinuert tid / analogt

Forskellige måder at definere integration i diskret tid / digitalt.

- Marginalt stabile filtre – alle har poler på enhedscirkel
- 4 IIR-filter-typer [1, s. 371-373]
 - Rektangulær
 - All-pole, "leaky integrator" med $a_1 = 1$:
 - $y_{li}(n) = x(n) + a_1 y_{li}(n - 1)$
 - At $a_1 = 1$ giver marginal stabilitet.
 - Trapez
 - Simpson's
 - Tick's (ikke vist her)

Forskellige karakteristika ift.

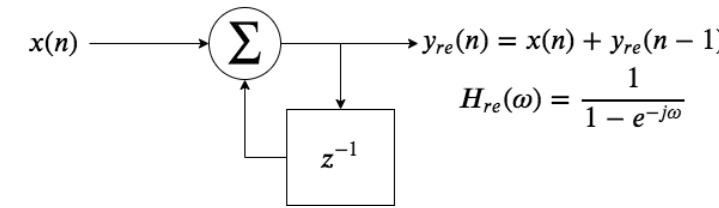
- Effektivt frekvensområde for input
- Frekvensrespons
- Støjimmunitet, støjdæmpning

$$H(\omega)e^{j\omega t} = \frac{1}{j\omega}e^{j\omega t}$$

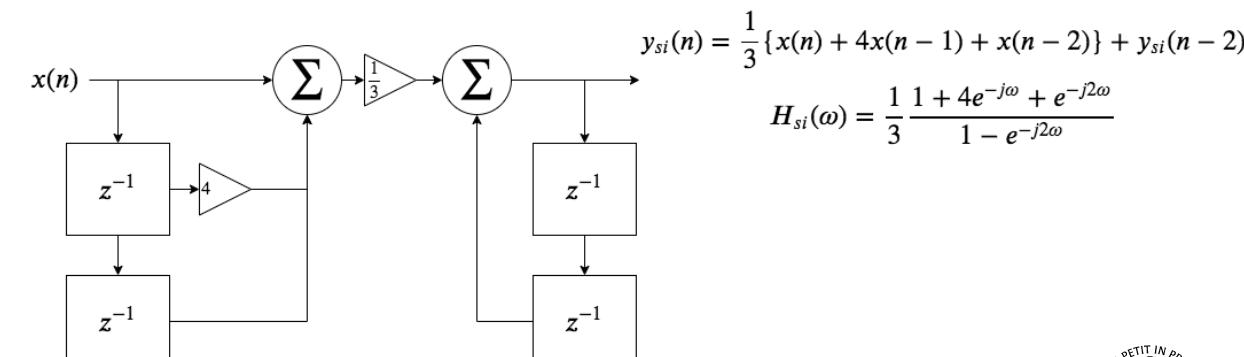
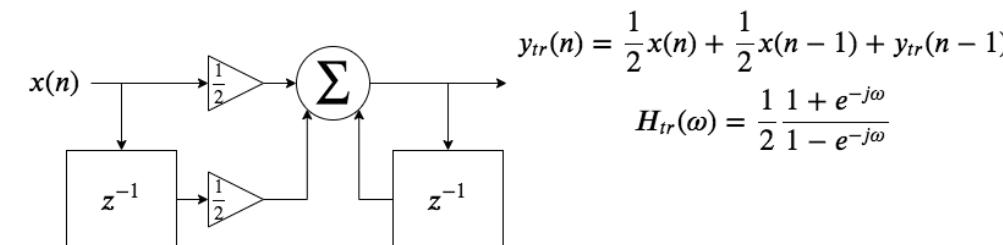
$$H(\omega) = \frac{1}{j\omega}$$

$$|H(\omega)| = \frac{1}{|\omega|}$$

$$\angle H(\omega) = -\text{sgn}(\omega)\frac{\pi}{2}$$



Bemærk, inverse operationer:
 $H_{re}(\omega) = \frac{1}{H_{fd}(\omega)}$

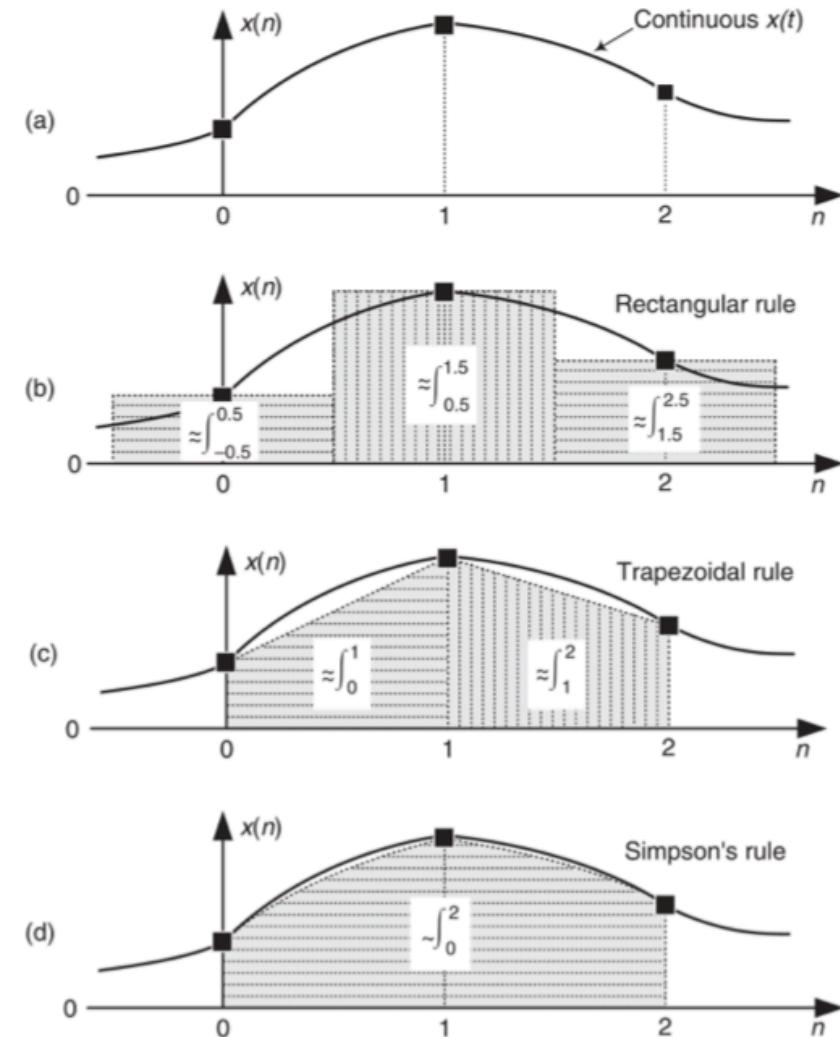


FORSTÅELSE AF INTEGRATORER VIA TIDSDOMÆNET

Fitter polynomier:

- Rektangulær
 - 0.-gradspoly fra 1 sample
 - Svarer til midtpunkts-Riemann
- Trapez
 - 1.-gradspoly fittet med 2 samples
 - Lineær interpolation
- Simpson's
 - 2.-gradspoly fittet med 3 samples
 - Kvadratisk interpolation

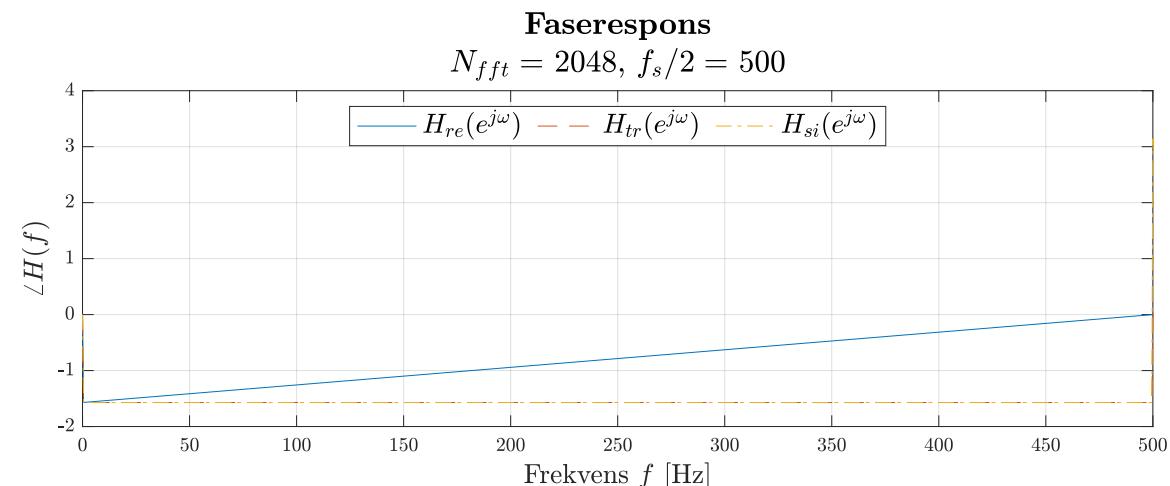
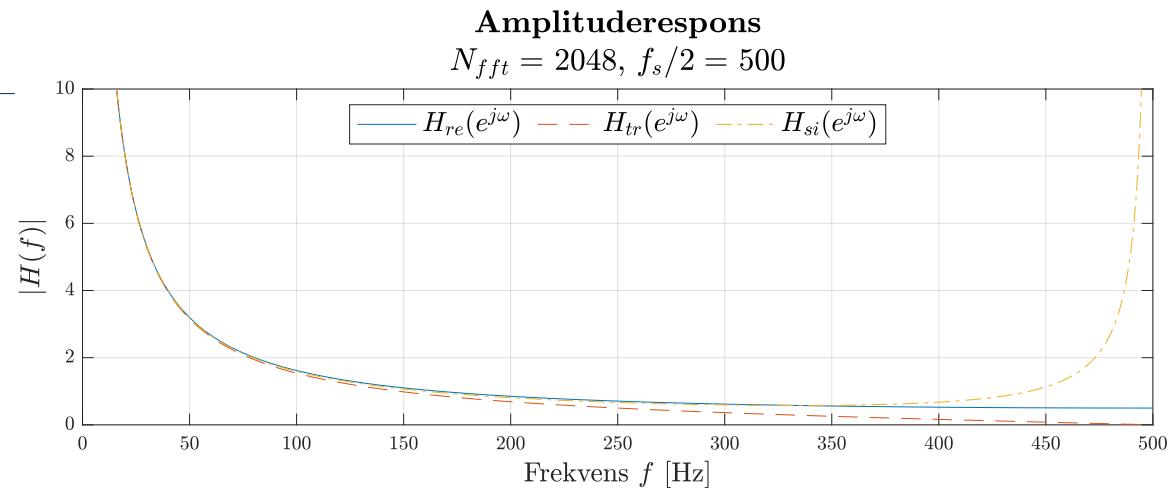
Principielt burde man kunne lave højere-ordens splines eller approksimationer (Taylor)



Kilde: [1, s. 371]

OVERBLIK: FILTERTYPER TIL DIGITAL INTEGRATION

Type	Fordele	Ulemper
Rektangulær	Dæmper højfrekvent støj Lineær fase	Ingen designparametre
Trapez	Dæmper højfrekvent støj lidt mere end rektangulær-filter Lineær fase	Ingen designparametre
Simpson's	Burde lave bedre approksimation <ul style="list-style-type: none"> Trade-off afhænger af data Lineær fase	Forstærker højfrekvent støj Mere kompleks beregning



CASE: TRE TYPER

Fysisk model: Omvendt situation af differentiation

- Integrerer målt hastighed, $v(n)$, op til estimeret position $x(n)$
- Additiv støj (wideband-støj, gaussisk hvidstøj)
- Hvis $v(t) = k \rightarrow x(t) = \int k dt = kt$ (rampe):
- Diskret tid: $x_{re}(n) = \sum_0^n kT_s = nkT_s$
- Hver sample af $v(n)$ repræsenterer T_s [sek.]

Filterdesign:

- Koefficienter aflæses direkte fra flow-diagram eller ofvf.fkt.
 - Rektangulær, 1. orden IIR: $b_{re} = 1, a_{re}(n) = [1, -1]$
 - Trapez, 1. orden IIR: $b_{tr}(n) = \frac{1}{2}[1, 1], a_{tr}(n) = [1, -1]$
 - Simpson's, 2. orden IIR: $b_{si}(n) = \frac{1}{2}[1, 4, 1], a_{si}(n) = [1, 0, -1]$

Konklusioner:

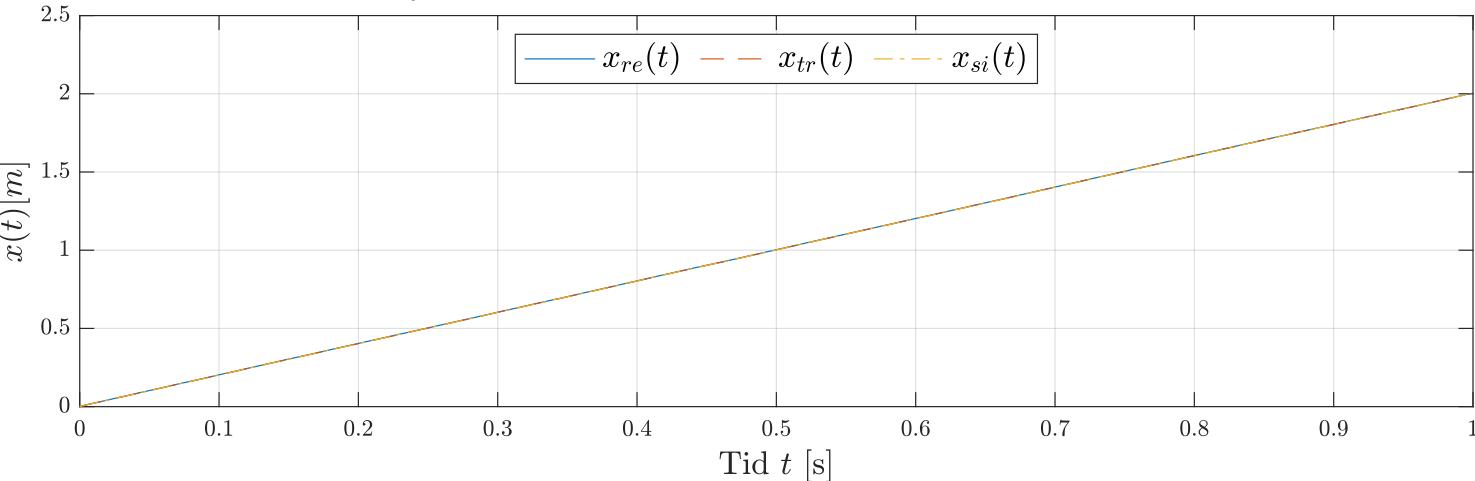
- Alle tre filtre performer fint, trapezoid-rule filteret er marginalt bedst til denne lineære data.
- Filtrene fungerer som LP
 - LF-støj passerer og integreres op (dc offset, LF)
 - Midband-støj dæmpes
 - HF-støj afhængig af filter
- Marginalt-stabile filtre.

Integration

$$x(t) = \int_0^t v(\tau) d\tau$$

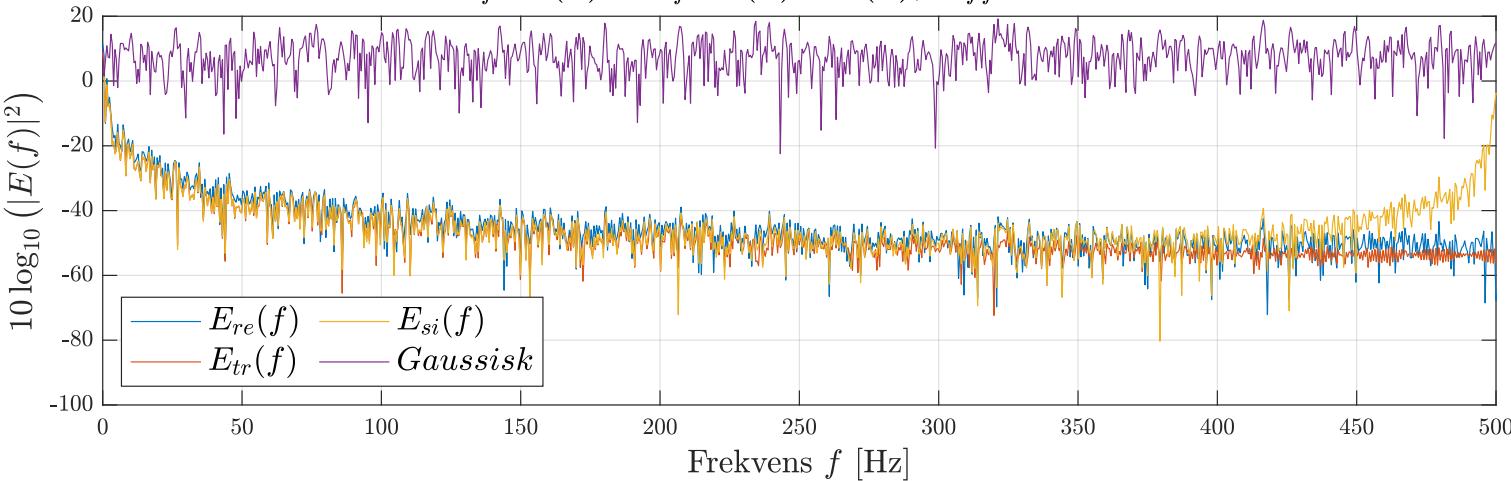
$$v(t) = 2 + \epsilon(t), \epsilon \sim N(0, \sigma^2), \sigma = 0.1$$

Absolut fejl i slutposition.: $x_{re} = 0.25\%$, $x_{tr} = 0.2\%$, $x_{si} = 0.28\%$



Effektspektrum for fejl

$$e_{filter}(n) = x_{filter}(n) - x(n), N_{fft} = 2048$$



OPSUMMERING

Præsentation af teori for differentiation og integration i det digitale domæne

Forskellige filertyper bør vælges til forskellige formål

- Kend din data!
- Frekvens i data?
- Behov for at justere designparametre, fx knækfrekvens?
- Forskellige grader af støjimmunitet / støjdæmpning?
- Behov for simplicitet i netværk / beregninger?

KILDEHENVISNING

- [1]: Richard G. Lyons. *Understanding Digital Signal Processing*. Prentice Hall. 2010.
- [2]: Dimitris G. Manolakis og Vinay K. Ingle. *Applied Digital Signal Processing: Theory and Practice*. Første udg. Cambridge University Press. 2011



AARHUS
UNIVERSITET