

### DISPOSITION 1 - DFT og Spektrogram

$$X_k = \frac{1}{N} \sum_{n=0}^{N-1} x_n e^{i2\pi k \frac{n}{N}}$$

To find the energy at a particular frequency, spin your signal around a circle at that frequency, and average a bunch of points along that path.

A few notes:

- $N$  = number of time samples we have
- $n$  = current sample we're considering (0 ..  $N-1$ )
- $x_n$  = value of the signal at time  $n$
- $k$  = current frequency we're considering (0 Hertz up to  $N-1$  Hertz)
- $X_k$  = amount of frequency  $k$  in the signal (amplitude and phase, a complex number)
- The  $1/N$  factor is usually moved to the *reverse transform* (going from frequencies back to time). This is allowed, though I prefer  $1/N$  in the forward transform since it gives the *actual* sizes for the time spikes. You can get wild and even use  $1/\sqrt{N}$  on both transforms (going forward and back still has the  $1/N$  factor).
- $n/N$  is the percent of the time we've gone through.  $2 * \pi * k$  is our speed in radians / sec.  $e^{-ix}$  is our backwards-moving circular path. The combination is how far we've moved, for this speed and time.
- The raw equations for the Fourier Transform just say "add the complex numbers". Many programming languages cannot handle complex numbers directly, so you convert everything to rectangular coordinates and add those.

### DISPOSITION 2 - Spektral forbreddning, zero padding og window i fht. DFT

Når man Fourier-transformerer et LTI  $s(t)$ -signal, er der altid spektral forbreddning (lækage), fordi det er umuligt at repræsentere alle indeholdte frekvenser i frekvensdomænet. Det der reelt sker i transformationen er, at vi ganger et vindue på det oprindelige signal. At gange et vindue på i det ene domæne (kontinuert tid / diskret tid) svarer til en foldning i det andet domæne. Derfor kan vi ligeså godt mindske spektral forbreddning ved at anvende f.eks. Hanning-vinduet. Dette vindue laver en vægtet sum af input-signalet, sådan at effekten af diskontinuerte funktioner (ændringer i frekvens) mindskes. Sidebåndene (sidelobes) kommer af den pludselig ændring mellem 0 og 1 (fra firkant-funktion). Denne pludselige ændring kan altså minimeres ved brug af vinduesfunktioner som f.eks. Hanning-vinduet, og dermed mindskes den spektrale forbreddning.

Brugen af vinduer (forskellig fra firkant-funktionen) mindsker dog amplituden af main-loben til omtrent halv størrelse. Dette resulterer i, at signaler med lav amplitude (der ligger tæt på main-loben) bliver nemmere at detektere, fordi de bliver mere synlige sammenlignet med main-loben.

Vinduesfunktionen kan derfor også bruges til at vise støjgulvet for et signal mere tydeligt, da sidesøjlerne dæmpes, og støjgulvet dermed bliver mere tydeligt.

Den dårlige ting ved vinduesfunktionen er at hovedsøjlen bliver en smule mere bred end for det oprindelige signal, hvilket gør det sværere at skelne signaler der ligger meget tæt.

The term 'leakage' usually refers to the effect of *windowing*, which is the product of  $s(t)$  with a different kind of function, the [window function](https://en.wikipedia.org/wiki/Window_function). Window functions happen to have finite duration, but that is not necessary to create leakage. Multiplication by a time-variant function is sufficient.

When both sampling and windowing are applied to  $s(t)$ , in either order, the leakage caused by windowing is a relatively localized spreading of frequency components, with often a blurring effect, whereas the aliasing caused by sampling is a periodic repetition of the entire blurred spectrum.

Fra <[https://en.wikipedia.org/wiki/Spectral\\_leakage](https://en.wikipedia.org/wiki/Spectral_leakage)>

### DISPOSITION 3 - FIR/IIR Filter Analyse + Design vha. pol- nulpunktsdiagram

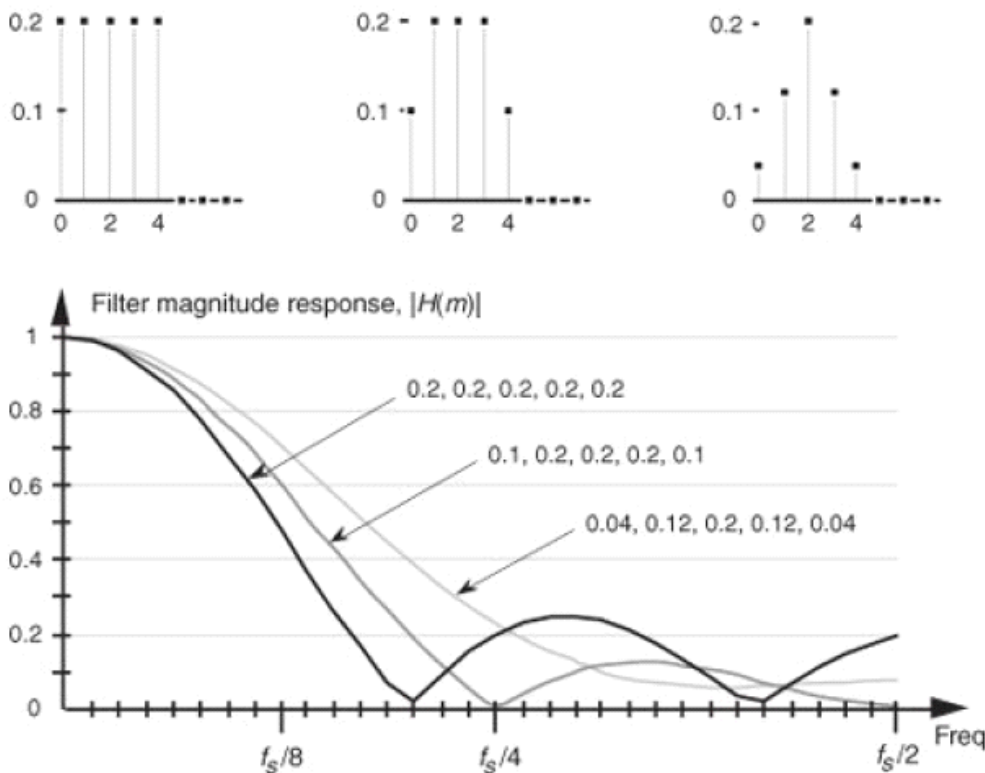
Der er to ting som beskriver et FIR filters frekvens response. Antal af taps (filter koefficienter) og størrelsen af disse koefficienter.  
Facts:

- For FIR filtre, vil dets impulse response og filter koefficienter være identiske.
- Foldning i tidsdomænet er det samme som multiplikation i frekvensdomænet og omvendt.

$$(5-8)$$
$$y(n) = h(k) * x(n) \xrightleftharpoons[\text{IDFT}]{\text{DFT}} H(m) \cdot X(m).$$

DFT af filter koefficienter er lig med filterets frekvensrespons. Der kan zeropaddes for at opnå større præcision.

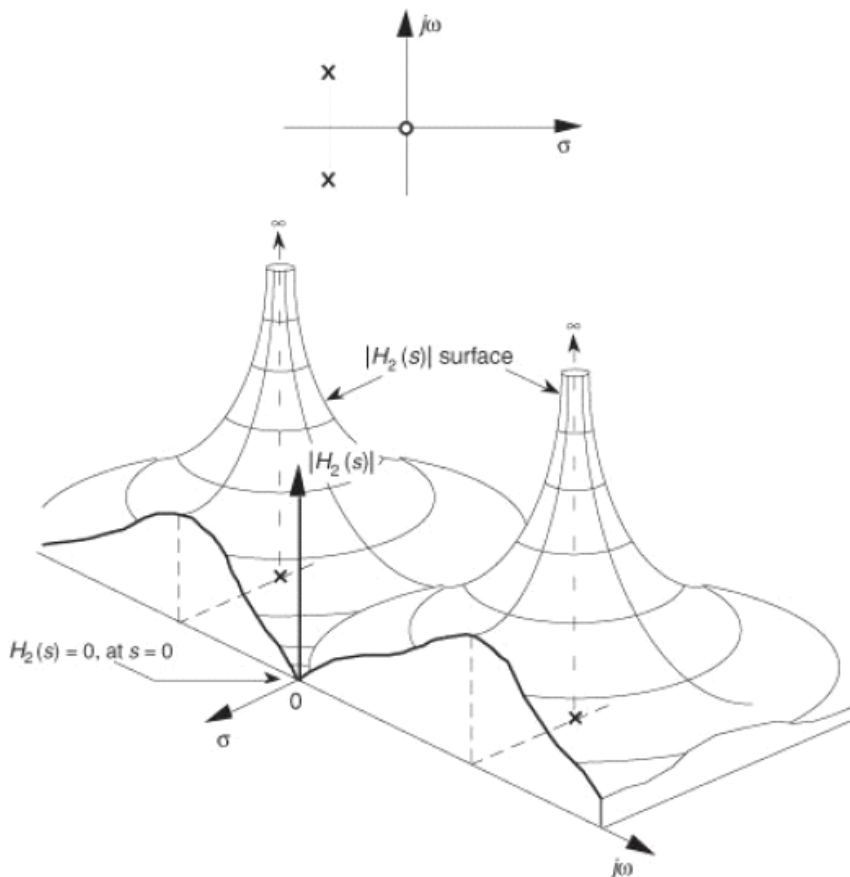
Et FIR filters indsvingningsforløb (transientresponse) er lig med antal koefficienter (unitdelay elements), hvorefter filterets *steady-state response* starter. Ligesom DFT på signaler, giver store spring i filterkoefficient-værdier mere flade sidelobes i frekvensresponsen, ligesom main lobe mere flad. Høj filterorden kræves for at holde main lobe stejl og sidebånd dæmpede, men giver også større indsvingningstid.



#### Pol- Nulpunktsdiagram metoden

Af nedenstående figurer ses, at nulpunkter trækker frekvensen ned til nul i det punkt de ligger, mens poler trækker frekvensen op. Teoretisk set mod uendelig høj amplitude, ved stigende frekvens.

Jo større magnitude poler og nulpunkter har, jo større effekt har de op frekvensresponsen. Dvs. de bliver mere betydende jo tættere de ligger på enhedscirklen, og bliver dermed også tættere på at være ustabile. (implementerings-problematikker). Fasen afgør placeringen af hhv. polens top og nulpunktets bund. Jo større fase, jo højere frekvenser.



Q&A: Hvis vi anvender konjugerede pol- og nulpunktspar, kan vi så se bort fra aliaserede frekvenser? Dette leder os frem til *cirkustelt-analogien*, hvor fremgangsmåden er:

Placer konjugerede poler og nulpunkter efter hvilke frekvenser der skal dæmpes og forstærkes. Filteret er stabilt så længe poler er indenfor enhedscirklen.

Poler og nulpunkter konjugeres for at opnå symmetri, og dermed udlignes bidragene fra den komplekse matematik. Det resulterer i, at vi kun får reelle koefficienter. Det gør at fasen bliver lineær.

*Cirkustelt-analogien i matematik:*

Opstil faktoriseret form af filterets overføringsfunktion hvor  $z_n$  og  $p_n$  er komplekse tal som repræsenterer de enkelte poler og nulpunkter:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{(z - z_0)(z - z_0^*)}{(z - p_0)(z - p_0^*)}$$

Ekspander tæller og nævner til formen:

$$H(z) = \frac{z^2 + k_1 \cdot z + k_2}{z^2 + k_3 \cdot z + k_4}$$

Gang igennem med  $z^{-k}$  så alle  $z$ 'er repræsenterer et delay:

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

Isoler  $Y(z)$ :

$$Y(z) \cdot (1 + z^{-1} + z^{-2}) = X(z) \cdot (1 + z^{-1} + z^{-2})$$

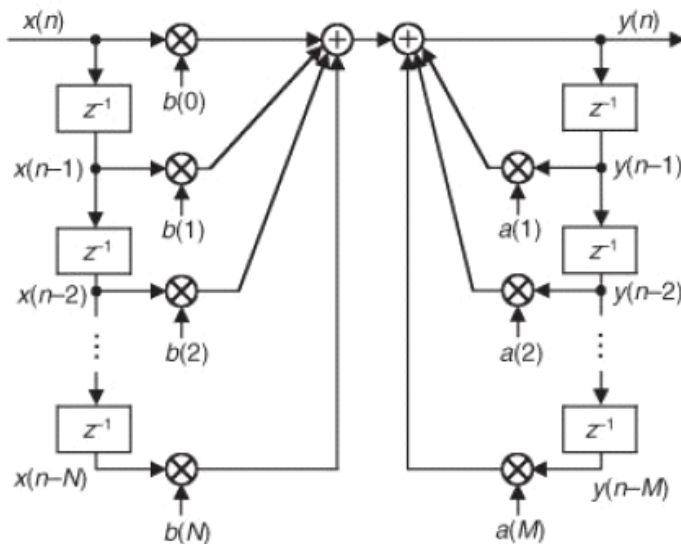
⇕

$$Y(z) = X(z) + X(z) \cdot z^{-1} + X(z) \cdot z^{-2} + Y(z) \cdot z^{-1} + Y(z) \cdot z^{-2}$$

Lav invers z-transformation for at opnå differensfunktionen, som bruges til implementering af filteret:

$$y(n) = b_0 \cdot x(n) + b_1 \cdot x(n-1) + b_2 \cdot x(n-2) + a_1 \cdot y(n-1) + a_2 \cdot y(n-2)$$

Implementeringen beskrives ofte med et signal flow diagram:



To conclude this section, we provide the following brief list of z-plane pole/zero properties that we should keep in mind as we work with digital filters:

- Filter poles are associated with increased frequency magnitude response (gain).
- Filter zeros are associated with decreased frequency magnitude response (attenuation).
- To be unconditionally stable all filter poles must reside inside the unit circle.
- Filter zeros do not affect filter stability.
- The closer a pole (zero) is to the unit circle, the stronger will be its effect on the filter's gain (attenuation) at the frequency associated with the pole's (zero's) angle.
- A pole (zero) located on the unit circle produces infinite filter gain (attenuation).
- If a pole is at the same z-plane location as a zero, they cancel each other.
- Poles or zeros located at the origin of the z-plane do not affect the frequency response of the filter.
- Filters whose transfer function denominator (numerator) polynomial has real-valued coefficients have poles (zeros) located on the real z-plane axis, or pairs of poles (zeros) that are complex conjugates of each other.
- For an FIR filter (transfer function denominator is unity) to have linear phase, any zero on the z-plane located at  $z_0 = Me^{j\alpha}$ , where  $z_0$  is not on the unit circle and  $\alpha$  is not zero, must be accompanied by a reciprocal zero whose location is  $1/z_0 = e^{-j\alpha}/M$ .
- What the last two bullets mean is that if an FIR filter has real-valued coefficients, is linear phase, and has a z-plane zero not located on the real z-plane axis or on the unit circle, that z-plane zero is a member of a "gang of four" zeros. If we know the z-plane location of one of those four zeros, then we know the location of the other three zeros.

#### **DISPOSITION 4 - Window method til FIR filter design**

Det første der skal udarbejdes i designet af et FIR filter, er det ønskede frekvens response. Dette betyder ideelt set (continuous filter) – hvis der designes et lavpas filter – at forstærkningen ved lave frekvenser er unity og efter en defineret cutoff frekvens vil dæmpningen vokse imod uendelig.

Der er dog én ting man skal tænke på, ved designet af et digitalt diskrete filter, at filter responset er periodisk med perioden,  $f_s$ .

Hvis man vil lave et FIR-filter med lineær fase er delayet givet ved:

$$D = \frac{\text{order}}{2} = \frac{N_{\text{taps}} - 1}{2} \text{ samples}$$

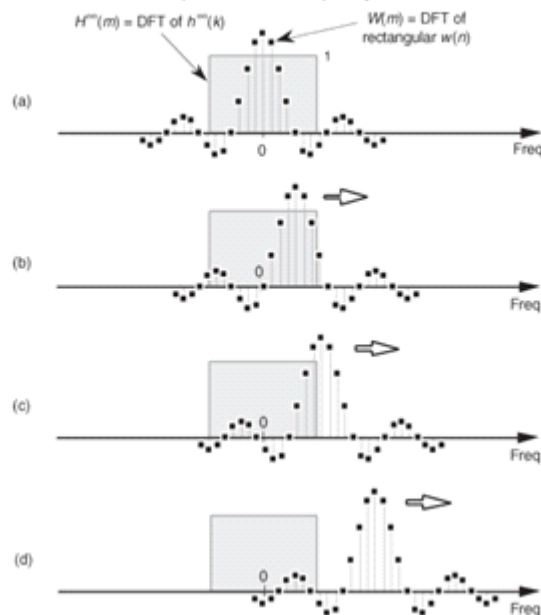
Derfor er man nødt til at have et ulige antal filter koefficienter, hvis man ønsker at delayet er lineært. Hvis man anvender et lige antal koefficienter, bliver responset ikke symmetrisk, og det en ulienær fase.

Jo flere filter-koefficienter man har, jo tættere kommer man på et ideelt filter, men indsvingningstiden stiger samtidig.

I teorien er et uendelig langt filter ideelt, men reelt set begrænses filter-længden, og dette giver magnitude ripples, som illustreret i figuren nedenfor. Det skyldes at DFT'en foldes med filteret, og filter-længden tager ikke højde for alle punkterne i DFT'en. Dette giver ripples.

Der sker altså en foldning af sinc-funktionen (rektangulære vindue i tidsdomæne) og et ideelt filter (firkant i frekvensdomæne).

**Figure 5-21 Convolution  $W(m) \cdot H^*(m)$ :** (a) unshifted  $W(m)$  and  $H^*(m)$ ; (b) shift of  $W(m)$  leading to ripples within  $H(m)$ 's positive-frequency passband; (c) shift of  $W(m)$  causing response roll-off near  $H(m)$ 's positive cutoff frequency; (d) shift of  $W(m)$  causing ripples beyond  $H(m)$ 's positive cutoff frequency.



Fun Fact:

Sinc-funktionen i tidsdomænet resulterer i et lavpasfilter i frekvensdomænet.

#### **DISPOSITION 5 - Interpolation og decimation**

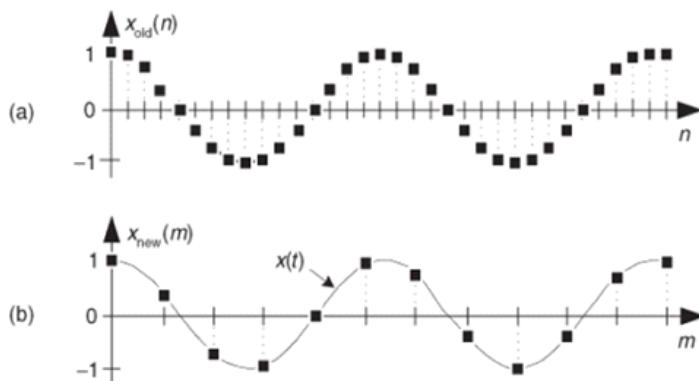
Tilsammen kaldes interpolation og decimation (op- og nedsampling) for resampling. Da man altid interpolerer eller decimerer med heltal, vil man ved at anvende både op- og nedsampling kunne ændre sample-frekvensen til hvilken som helst ønsket størrelse, da de to heltal derfor vil indgå i en samlet brøk, der definerer den nye samplefrekvens.

Principperne bruges også ofte når 2 eller flere forskellige systemer sender data til hinanden med forskellige samplingshastigheder. Disse skal passe sammen for at udvekslingen foregår optimalt.

#### **Decimation**

Herunder vises et eksempel på nedsampling med en faktor 3. Vi beholder altså kun 1/3 af de oprindelige samples.

**Figure 10-1 Sample rate conversion:** (a) original sequence; (b) downsampled by  $M = 3$  sequence



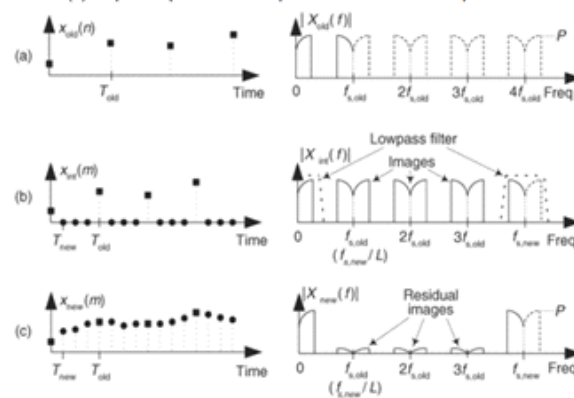
Da en decimation rykker aliaserede frekvenser tættere sammen, vil man i praksis altid lavpasfiltrere før en decimation.

Q&A: FØR en decimation??

#### **Interpolation**

Herunder vises et eksempel hvor samplefrekvensen ønskes forøget med en faktor 4. I realiteten indsættes tre samples med værdien 0. Disse værdier ændres idet man anvender et lavpasfilter (interpolationsfilter), der midler over tid, og dermed interpolerer (indsætter?) de midlede værdier. Dermed øges samplefrekvensen. Dette resulterer i at der dannes aliaserede replika omkring heltal af den gamle sample-frekvens. Disse mindskes også vha. interpolationsfilteret.

**Figure 10-7** Interpolation by four: (a) original sampled sequence and its spectrum; (b) zeros inserted in original sequence and resulting spectrum; (c) output sequence of interpolation filter and final spectrum.



Q&A: Forskellige måder at interpolere på?

#### DISPOSITION 6 - Differentiation og Integration

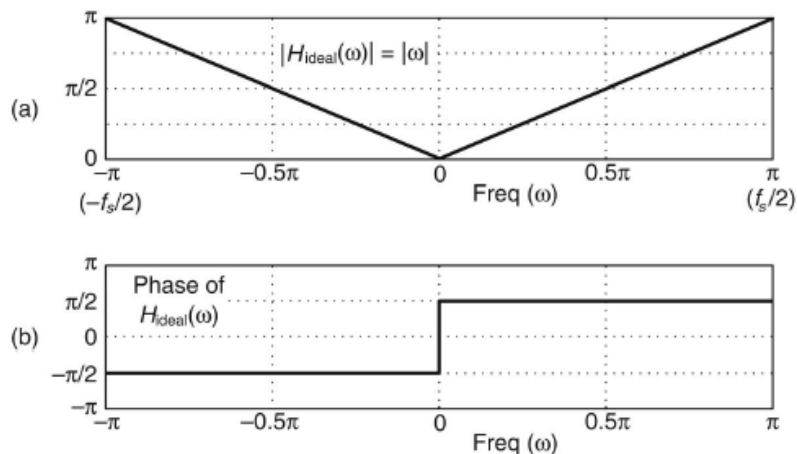
*Differentiators:*

Anvendes til at differentiere et signal, til f.eks. demodulation af FM signaler.

Husk at 's' svarer til differentiation i frekvensdomænet. Dette medfører:

$$H(s) = s \rightarrow H(j\omega) = j\omega$$

Det giver følgende amplitude- og fasekarakteristik:



Af disse karakteristikkere ses det, at for negative værdier, falder amplituden ved større frekvenser, men når vi har positive værdier, stiger amplituden når vi bevæger os mod højere frekvenser. Dette fænomen ses også hvis vi f.eks. differentierer en sinus:

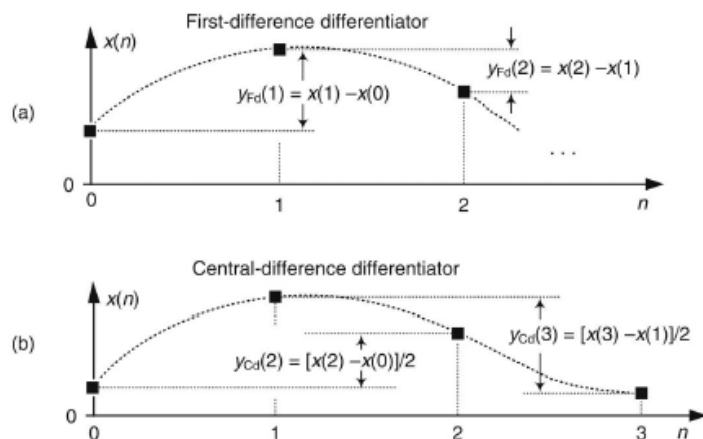
$$\frac{d}{dt} \sin(\omega t) = \omega \cdot \cos(\omega t)$$

Heraf ses det at amplituden vokser lineært med  $\omega$  når frekvensen stiger (for positive frekvenser).

Differentiation kan implementeres på to måder.

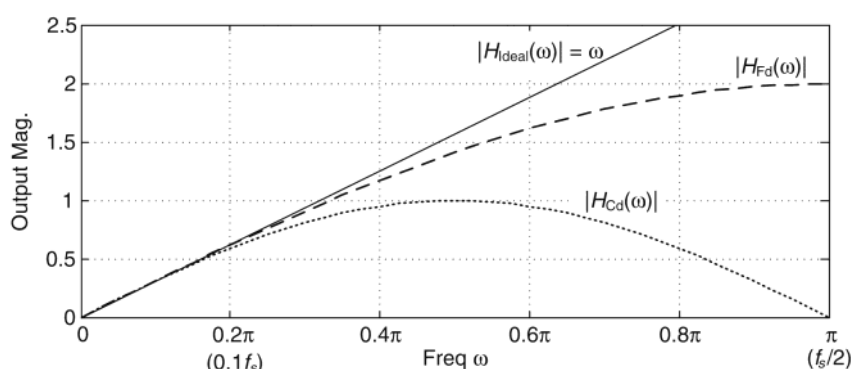
First-difference	$y_{Fd}(n) = x(n) - x(n-1)$
Central-difference	$y_{Cd}(n) = (x(n) - x(n-2))/2$

De to funktioner ses afbildet nedenfor.



Heraf ses selve implementeringen på et signal. For first-difference bestemmes hældningen af kurven som forskellen på den nuværende og den tidligere bin. Mens central-difference differentierer (midler) over en ekstra sample.

Både den ideelle, first-difference og central-differences amplitudekarakteristik ses afbildet nedenfor.



Heraf ses det, at First-difference forstærker en større del af høj-frekvent støj, hvilket central-difference er bedre til at dæmpe. First-difference er derimod en lineært frekvensafhængig op til højere frekvenser end ved central-difference (båndbredde for differentiatoren). Pros and cons...

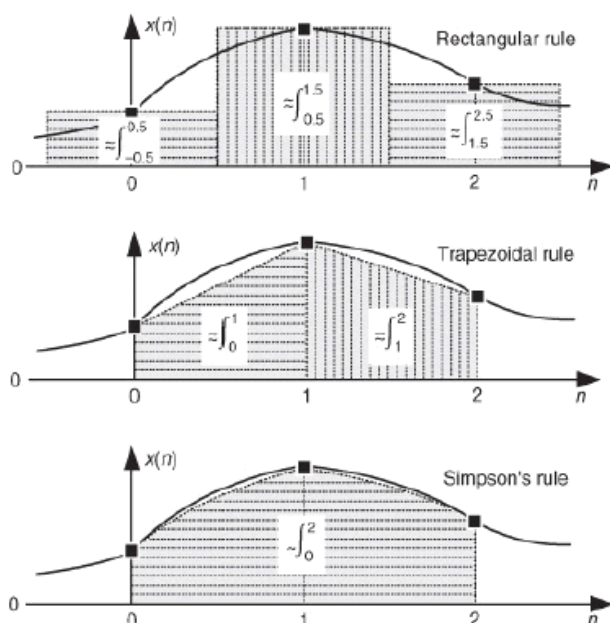
Endnu et vigtigt element er den lineære fase. Denne fås som sagt for FIR-filters, hvis antallet af filter-koefficienter er anti-symmetriske, hvilket vil betyde at de har et konstant tidsdelay (group delay) som er givet ved:

$$G_{diff} = \frac{D}{2} \text{ samples}$$

Hvor D er antallet af unit-delays i filtreringen. Det er i den forbindelse meget vigtigt hvis flere systemer skal synkroniseres, at deres  $G_{diff}$  (group delay) er en integer (helt tal). Her vil first-difference differentiatoren have ét unit-delay og altså ende med  $G_{diff} = \frac{1}{2} \text{ samples}$ , hvilket IKKE er en integer, hvor central-difference differentiator vil have et group delay på  $G_{diff} = \frac{2}{2} = 1 \text{ samples}$ .

#### Integration

Integration er veldefineret i det analoge domæne og for at approksimere integration i det digitale domæne, bruger vi filtre, der laver numerisk integration af samplede signaler. Digital integration kan bl.a. gøres efter følgende regler:

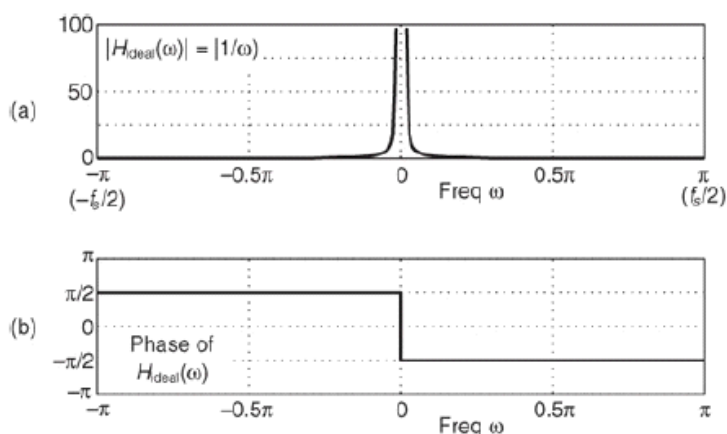


Regel	Differensligning	Forklaring
Rektangel	$y_{Re}(n) = x(n) + y_{Re}(n-1)$	Rektangulær sum omkring $\pm 1/2$ af sample
Trapezoid	$y_{Tr}(n) = 0.5x(n) + 0.5x(n-1) + y_{Tr}(n-1)$	Gennemsnit af nuværende og tidligere sample
Simpson's	$y_{Si}(n) = [x(n) + 4x(n-1) + x(n-2)]/[3 + y_{Si}(n-2)]$	3 samples approkimerer en anden ordens funktion under kurven, med toppunkt i det midterste sample.
Tick's	$y_{Ti}(n) = 0.3584x(n) + 1.2832x(n-1) + 0.3584x(n-2) + y_{Ti}(n-2)$	Designet til at være præcis ved lave frekvenser $0 \leq \omega \leq \pi/2$ radianer/sample ( $0 \rightarrow fs/4$ Hz), på bekostning af præcision ved højere frekvenser.

Ovenstående digitale integratorers sammenlignes med en analog integrator. Husk at cosinus integreret giver sinus over omega:

$$\int \cos(\omega t) dt = \frac{\sin(\omega t)}{\omega}$$

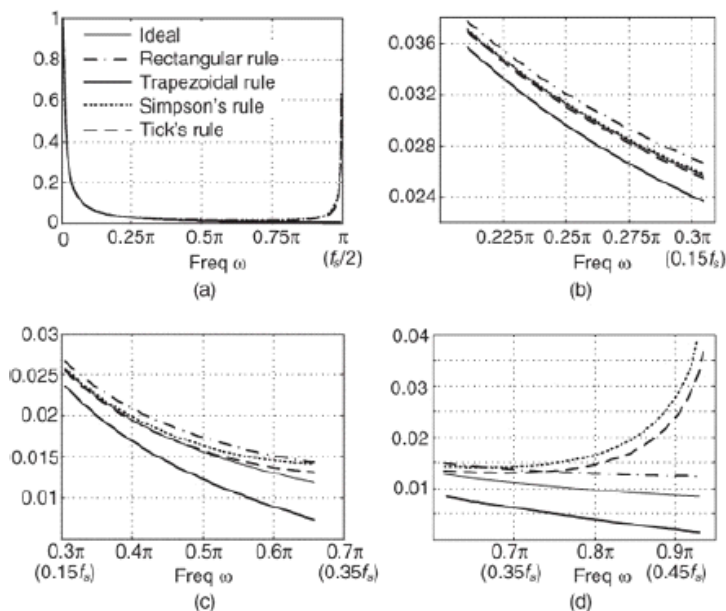
Heraf ses det, at en sinusfunktion faseskiftes under integration med  $\pi/2$  ( $-90^\circ$ ), og dæmpes med en faktor  $1/\omega$ .  
Dermed er amplituderesponsen af en ideel integrator:  $|H_{ideal}| = |1/\omega|$ . Amplitude- og faserespons er afbildet nedenfor:



Heraf ses det, at amplituden vokser mod uendelig, når frekvensen går mod 0. Fasedrejet ses også.

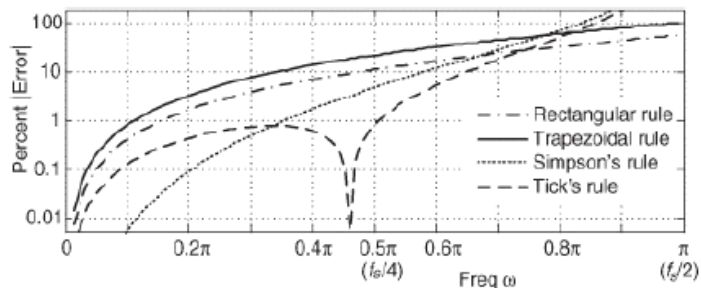
Af nedenstående fire grafer ses amplituderesponsen af de fire digitale integratorer sammenlignet med en ideel integrator. De fire grafer viser amplituderesponsen ved forskellige båndbredder (i det positive frekvensområde). Graferne er normaliserede.





På figur (b) ligger Tick's og Simpson's oveni den ideelle respons.

Det er interessant at kigge på den procentvise fejl af de fire integratorer, i forhold til den ideelle:



Heraf ses det, at fejlen er meget lav ved lave frekvenser for alle integratorerne. Dette er en vigtig egenskab, da det er her forstærkningen af integratorerne er størst, som det ses på figur (a) ovenfor. Derfor er fejlen mindst ved lave frekvenser.

Med lave frekvenser menes, at de spektrale komponenter af det signal der skal integreres er lave i frekvens, relativt til  $f_s$ .

Der er dog problemer med Simpson's og Tick's integratorerne ved frekvenser nær  $f_s/2$  Hz, fordi de forstærker højfrekvent støj i området tæt på  $f_s/2$  Hz. Det skyldes at disse integrators overføringsfunktion har poler i  $z = -1$  (uendelig forstærkning), svarende til en cyklisk frekvens af  $f_s/2$  Hz. I scenarier med højfrekvent støj bør man derfor anvende rektangulær eller trapezoid integratorerne, da disse har forbedret forstærkning i nærheden af  $f_s/2$  Hz.

De ovenstående integratorer er rekursive netværk, der samtidig har lineær fase. Der er dog kun trapezoid, Simpson's og Tick's der har ideelt faserespons. Der kan opstå problemer med offset-fejl og drift, fordi en integration her vil vokse mod uendelig.

Alle de ovenstående integratorer har poler i  $z = 1$ , svarende til en cyklisk frekvens på 0 Hz. Disse poler medfører at hardware registre i integratoren kan overflowe, hvis input-sekvensen  $x(n)$  har et amplitude offset. Registerne skal derfor være store nok til at garantere at offsettet ikke resulterer i et overflow, som giver korrupt data i integratoren.

#### **DISPOSITION 7 - Stokastiske signaler (Middelværdi, varians, sandsynlighedstæthedsfunktion og histogram)**

*Middelværdi*

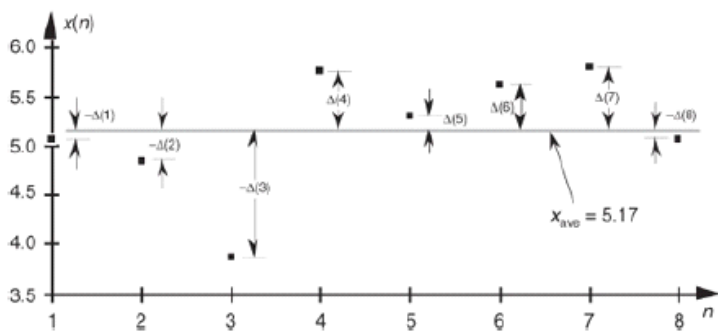
$$x_{ave} = \frac{1}{N} \cdot \sum_{n=0}^{N-1} x(n) = \frac{x(1) + x(2) + \dots + x(N)}{N}$$

Bestemmelse af hvad støj kan ske ved at midle støjsignaler, fordi middelværdien for de enkelte komponenter ikke ændrer sig.

- Middelværdien er blot DC-værdien.
- Standardafvigelsen (spredningen) siger noget om amplituden i analyse-signalet. 67% af værdierne ligger inden for  $\pm$  denne værdi, som resultat af sandsynlighedsfunktionen (normalfordelingen).
- Variansen er i realiteten altid et estimat (indikeres med en  $\wedge$ ).

$$\widehat{\sigma^2} = \frac{1}{N} \cdot \sum_{n=0}^{N-1} (x(n) - x_{ave})^2 = \frac{(x(1) - x_{ave})^2 + (x(2) - x_{ave})^2 + \dots + (x(N) - x_{ave})^2}{N}$$

Det er et udtryk for hvor meget hver enkelt værdi fluktuerer (afviger) fra middelværdien. Denne værdi siger f.eks. noget om kvantiseringsfejl og er anvendelig til brug i midlings-algoritmer. Variansen er grafisk afbildet i nedenstående graf.



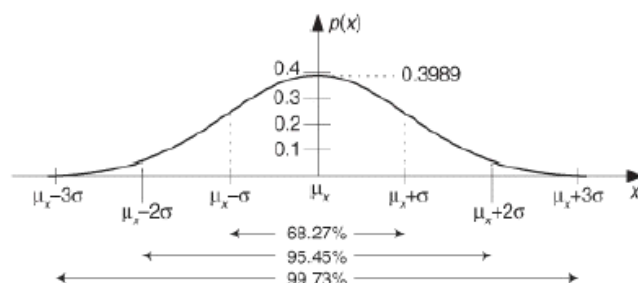
### Probability Density Function (PDF)

Sandsynlighedstæthedsfunktionen er defineret som:

$$p(x) = \frac{1}{\sigma \cdot \sqrt{2\pi}} \cdot e^{-(x-\mu_x)^2 / 2\sigma^2}$$

Funktionen kaldes også Normal PDF og Gaussisk PDF. Fordelingen egner sig især godt til at teste software algoritmer og hardware processorer.

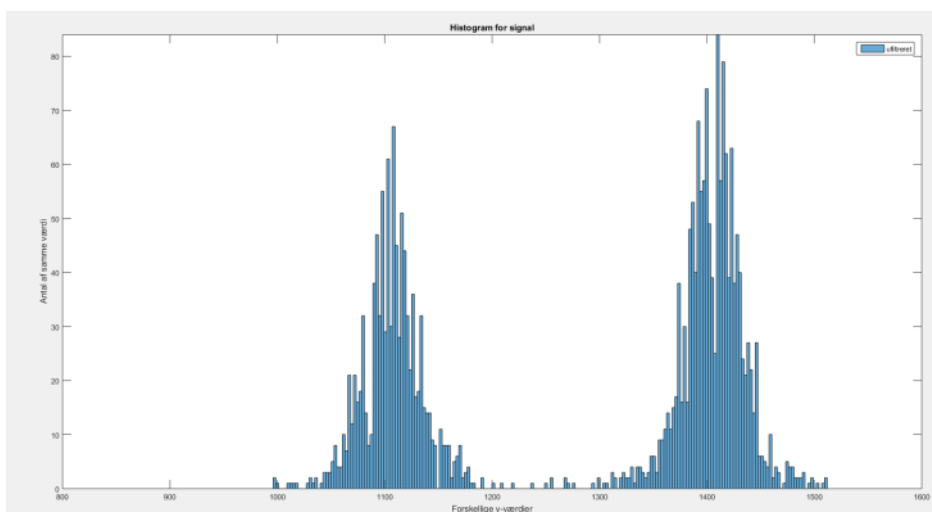
**Figure D-8** A normal PDF with mean =  $\mu_x$  and standard deviation =  $\sigma$ .



Arealet under kurven er altid 1. Med procenterne under grafen indikeres, hvordan input-data fordeles procentvist (hvis funktionen har en normal distribution). 68,27% af data vil være fordelt indenfor værdien af standard afvigelsen  $\sigma$ .

### Histogram

I Case 3: Vejecelle lavede vi et histogram over måledata for en vejecelle.



Heraf ses det, at de to grafer er nogenlunde normalfordelt omkring middelværdien.

### DISPOSITION 8 - Beregning af SNR i tids- og frekvensdomænet

Til beregning af average power for diskrete samples, defineres rms-værdien som:

$$x_{\text{rms}} = \sqrt{\frac{1}{N} \sum_{n=1}^N x(n)^2}$$

For en kontinuert sinus  $x(t) = A_p \sin(\omega t)$  med middelværdi 0, er rms-værdien defineret som:

$$x_{rms\ sine} = \frac{A_p}{\sqrt{2}}$$

Når et signal har en middelværdi på 0, er dets rms-værdi lig med signalets standard afvigelse.

### SNR

En måde at kvantificere kvaliteten af et støjfyldt signal. SNR er ratioen mellem effekten af det støjfri signal og selve støjen:

$$SNR = \frac{\text{Signal power}}{\text{Noise power}}$$

### Tidsdomænet

Hvis vi har et signal  $x_s(n)$  og nogle støj samples  $x_n(n)$ , kan vi beregne SNR for det samlede signal  $x(n) = x_s(n) + x_n(n)$ :

$$SNR = \frac{\text{Signal power}}{\text{Noise power}} = \frac{\frac{1}{N} \sum_{n=0}^{N-1} [x_s(n)]^2}{\frac{1}{N} \sum_{n=0}^{N-1} [x_n(n)]^2}$$

Midlingen i både tæller og nævner ( $1/N$ ) er medtaget for forståelsens skyld, men vil selvfølgelig gå ud med hinanden.

Hvis man kender variansen af både signalet og støjen, kan vi udtrykke SNR for AC-delen af signalet som:

$$SNR = \frac{\text{Signal variance}}{\text{Noise variance}} = \frac{\sigma_s^2}{\sigma_n^2}$$

Effekt udtrykkes både i meget store og meget små værdier (militær radar = Megawatt, Mobiltelefon = microwatt).

Derfor udtrykkes SNR normalt på logaritmisk skala (dB):

$$SNR_{dB} = 10 \cdot \log_{10}(SNR) \text{ dB} = 20 \cdot \log_{10} \left( \frac{x_s(n)_{rms}}{x_n(n)_{rms}} \right) \text{ dB}$$

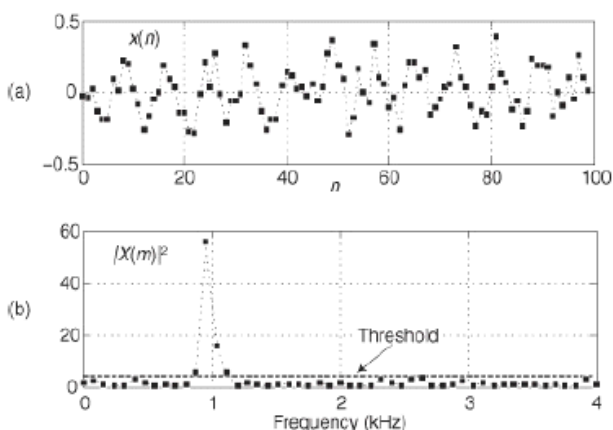
Hvis man kender standard afvigelsen af både signalet og støjen, kan vi udtrykke SNR for AC-delen af signalet som:

$$SNR_{dB} = 20 \cdot \log_{10} \left( \frac{\sigma_s}{\sigma_n} \right) \text{ dB}$$

### Frekvensdomænet

Standard-proceduren er at man laver en DFT på input-signalet (svarende til længden af signalet) med en samplefrekvens der er stor nok. Derefter beregnes energiindholdet og effekten af de samlede værdier. Disse afbildes som vist i figuren nedenfor.

The standard procedure for doing so is as follows: Assume we have  $N = 100$  samples of the noisy 986 Hz real-valued  $x(n)$  sinusoid, where the sample rate is  $f_s = 8$  kHz, as shown in Figure D-5(a). After performing a 100-point DFT, and computing the spectral magnitude-squared sample values, we obtain the positive-frequency  $|X(m)|^2$  power spectral samples depicted in Figure D-5(b).



I figur (a) ses det støjfyldte input-signal og i figur (b) det spektrale effekt-indhold efter DFT'en.

Derefter bestemmes en rimelig tærskel værdi (indikeret af den stiplede linje i figur (b)). Over denne værdi er der kun signal-relateret effekt, mens der under linjen kun er støj. Dermed kan SNR beregnes som summen af effekten af hhv. signal og støj.

$$SNR = \frac{\text{Sum of the } |X(m)|^2 \text{ samples above Threshold}}{\text{Sum of the } |X(m)|^2 \text{ samples below Threshold}}$$

Der er ting at bemærke mht. bestemmelse af SNR ud fra samples i frekvensdomænet:

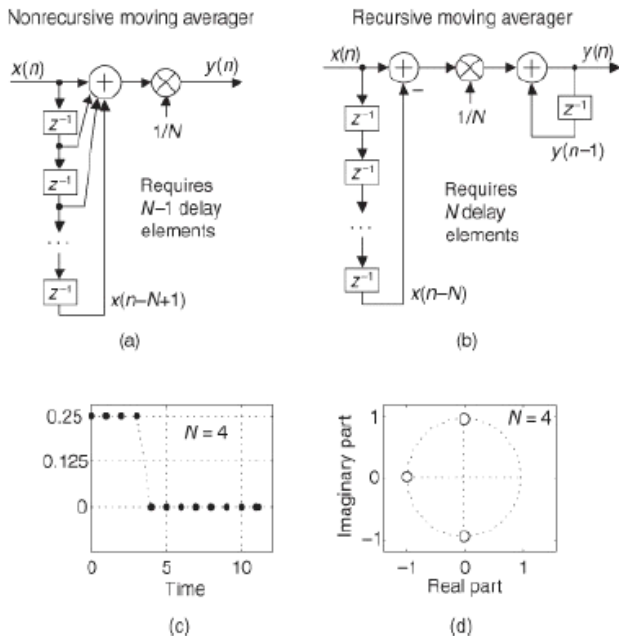
- Længden af inputtet bør være et kvadrattal, for at FFT kan anvendes til at bestemme effekten  $|X(m)|^2$  effektivt.
- Grundet spektral symmetri af reelle tidssignaler, behøver vi kun at bestemme  $|X(m)|^2$  for positive frekvenser.
- Tærskelværdien bør vælges så vi sikrer os at så meget som muligt af signalet ligger over denne værdi.

- Vi bør foretage FFT på mange sekvenser, for at bestemme mange sekvenser af  $|X(m)|$ . Disse sekvenser midles derefter, inden en endelig beregning af effekten  $|X(m)|^2$  foretages. Denne endelig værdi anvendes til at beregnes SNR. Dermed forbedres præcisionen (variansen mindskes) af vores SNR beregning.

• If we repeat our SNR estimation computation on multiple non-overlapping N-sample  $x(n)$  sequences, we'll see a noticeable variation (variance) in the various SNR estimation results. To improve the accuracy, and repeatability, of our SNR estimation it's prudent to collect many blocks of N-sample  $x(n)$  sequences and perform many FFTs to compute multiple  $|X(m)|$  magnitude sequences. Then those multiple  $|X(m)|$  sequences are averaged before computing a single  $|X(m)|^2$  power sequence for use in Eq. (D-19). The idea is to improve the accuracy (reduce the variance) of our SNR estimations by way of averaging as indicated by Eq. (D-2).

### DISPOSITION 9 - Midlingsfiltre

Vi vil gerne midle signaler for at øge præcisionen af dem. I et reeltids-system giver det ingen mening at sample et givent antal samples og bagefter midle over disse. Det ville resultere i at vi skulle vente et stykke tid før vi kunne midle igen. I stedet bruges digitale midlingsfiltre. Husk på at midling i tidsdomænet har lavpas-karakter (se figur (c), et 4.ordens Moving Average filter).



Vi ser at figur (a) og figur (b) er fuldstændig ens, bortset fra at (b) er en rekursiv udgave, og dermed meget mere effektiv (Der kræves kun to summationer for et filter af en given orden).

De to filtre har identiske impulsrespons og identisk lineær-fase frekvensrespons. Den rekursive udgave er blot en mere effektiv implementering af begrebet "N-punkt midlingsfilter".

De to filtre har dog den ulempe at de forsinker signalet, afhængigt af filterordenen. For det rekursive filter kan outputtets amplitude stige, og potentielt overflowe hardware registre.

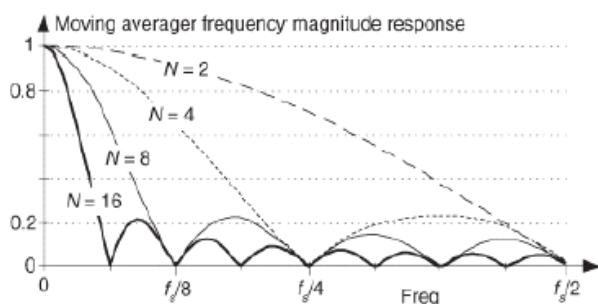
Hvis  $N$  er et kvadrattal, kan multiplikationen  $(1/N)$  implementeres som right shift i koden, hvilket er mere effektivt.

De to midlingsfiltre har amplituderensonsen:

$$|H_{ma}(f)| = \frac{1}{N} \left| \frac{\sin(\pi f N)}{\pi f N} \right|$$

Hvor den normaliserede frekvens  $f - 0.5 \leq f \leq 0.5$  svarer til det kontinuerte frekvensområde  $-f_s/2 \rightarrow f_s/2$  Hz

Nedenfor er afbildet amplituderensonsen for forskellige ordener af midlingsfiltrene:



De approksimeres som sinc-funktionen, fordi de svarer til midlingsfilterets rektangulære impulsrespons i tidsdomænet.

Idet  $N$  stiger bliver forstærkningen mere smal, og dermed forstærkes støj af input-signalet også mere.

Nulpunkterne ses at ligge ved givne frekvenser, som også kan ses ud fra pol-nulpunktsdiagrammet for filteret. Disse nulpunkter

vil være lokaliseret i følgende vinkler ( $\theta_{\text{zeros}}$ ) og have nul forstærkning ( $f_{\text{nulls}}$ ) i følgende frekvenser:

$$\theta_{\text{zeros}} = k \cdot \frac{2\pi}{N} \text{ radianer}, \quad f_{\text{nulls}} = k \cdot \frac{f_s}{N} \text{ Hz}, \quad \text{hvor } k = 1, 2, 3, \dots, N-1$$

En vigtig detalje er, at udgangseffekten (variansen) er givet ved indgangseffekten delt med filterordenen:

$$\sigma_{\text{out}}^2 = \frac{\sigma_{\text{in}}^2}{N}$$

Moving average filtre kommer til kort på effektiviteten, og i stedet kan det mere effektive eksponentielle midlingsfilter bruges.

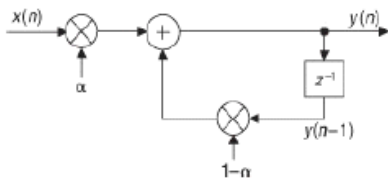
#### Ekspontielt midlingsfilter

Et simpelt rekursivt lavpasfilter, beskrevet ved følgende differensligning:

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

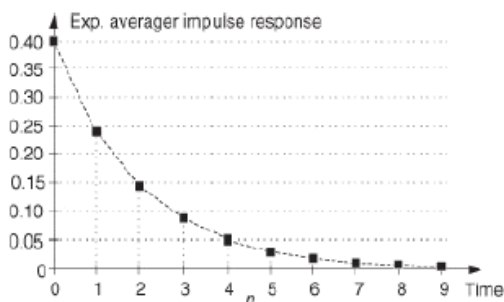
$\alpha$  bestemmer hvor godt filteret er. Kravet er:  $0 < \alpha < 1$ , og jo tættere  $\alpha$  er på 1, jo mindre orden og indsvingningstid får filteret. Det modsatte gælder selvfølgelig, jo tættere  $\alpha$ -værdien kommer på 0 (Lavere orden og indsvingningstid).

Det eksponentielle midlingsfilter implementeres som vist nedenfor:



Det ses at det eksponentielle midlingsfilter bruger færre beregninger per output sample end et standard ikke-rekursivt Moving Average filter. Derudover kræver det ikke særlig meget hukommelse at implementere (kun et enkelt delay element).

Herunder ses impulsresponsen af et exp. midlingsfilter i tidsdomænet ( $\alpha = 0.4$ ):



Hvis  $\alpha$  har værdien 1, vil filteret reagere instant på et givet input, men i takt med at  $\alpha$  bliver mindre begynder vægten fra den eksponentielle midling at betyde mere og mere. Dette vil udglatte støjfyldte signaler. Det er et trade-off mellem, hvor hurtigt filteret reagerer på inputtet, i forhold til hvor meget støjen dæmpes.

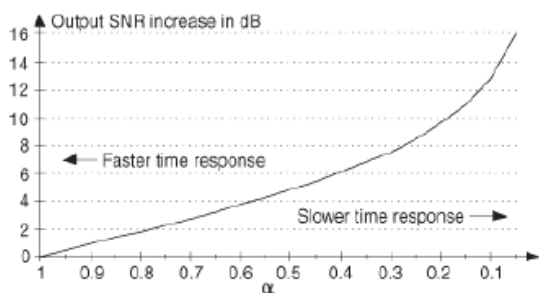
For et eksponentielt midlingsfilter gælder, at støjen kan reduceres med en faktor  $R$ , ved at vælge  $\alpha$  til at være:

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

SNR af det eksponentielle midlingsfilter kan hæves, fordi det sænker støjen af signalet. Forholdet er:

$$SNR_{\text{exp(dB)}} = 10 \cdot \log_{10} \left( \frac{\alpha}{2 - \alpha} \right)$$

Formlen afbildes:



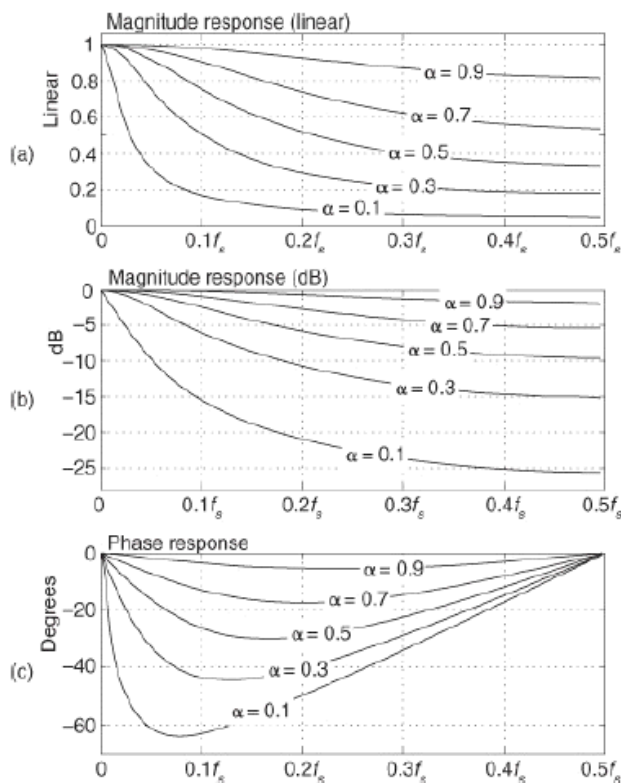
Heraf ses trade-off mellem støjreduktion og hurtighed af filterets respons.

#### Frekvensdomænet

Det eksponentielle midlingsfilter er et 1. ordens IIR-filter med z-domæne overføringsfunktionen:

$$H_{exp}(z) = \frac{\alpha}{1 - (1 - \alpha)z^{-1}}$$

Sætter vi  $z = e^{j\omega}$  kan frekvensresponsen opskrives, og den afbildes ( $0 \rightarrow \frac{f_s}{2}$ ) i nedenstående amplitude- og faseresponser.



Når  $\alpha$  reduceres i værdi, får vi et smallere pasbånd, som illustreret i figur (a)

Hvis man gerne vil designe et eksponentielt midlingfilter (som er afbildet i figur (b)) med en given knæfrekvens  $f_c$ , så kan man opstille en formel for den tilsvarende alpha-værdi:

$$\alpha = \cos(2f_c/f_s) - 1 + \sqrt{\cos^2(2f_c/f_s) - 4\cos(2f_c/f_s) + 3}$$

På trods af den ulinære fasekarakteristik, har det eksponentielle midlingsfilter ikke fasedrej i 0 Hz, og dermed passerer DC-signaler igennem uden at blive fasedrøjet.

En magnetkortlæser anvender et eksponentielt midlingsfilter. Aflæsningen af magnetstripen digitaliseres med en A/D konverter, og de diskrete samples midles eksponentielt før detektion af binær data foretages.

#### **DISPOSITION 10 - Auto- og krydskorrelation**

Kryds-korrelation er et udtryk for hvor meget et signal ligner sig selv, efter det er blevet forskudt et antal samples.

Vi kan bruge det til at finde ud af et bestemt signal i et input vi har målt, fordi den maksimale værdi af outputtet findes, i netop det punkt hvor det efterspurgte signal er til stede.

Det gøres ved at indsende et signal, måle det igen og korrelere disse to signaler med hinanden, resulterende i et tredje signal.

Selvom signalet indeholder negative samples, bliver outputtet af korrelationen altid positiv, da de to signaler (målt signal og korrelations-signal) ganges sammen.

X-korrelations-signalet er altid symmetrisk (selvom inputtet er asymmetrisk). Og bredden af den midterste peak er dobbelt så bred som peaken af input-signalet. Det kaldes også 'matched filtering'

F.eks. kan vi generere et hvidstøjssignal på 1000 samples, og lægge et kendt signal ind i det på 100 samples, som vil resultere i en maksimal værdi, ved det kendte signal, når vi kryds-korrelerer de to signaler.

Q&A: Når kryds-korrelationen er maksimal, er værdien lig variansen?

Da korrelation og foldning er ens på nær rækkefølgen de sker i, kan man beskrive korrelation mellem to signaler  $a(n)$  og  $b(n)$  som

$$a(n) * b(-n) = a(-n) * b(n)$$

Denne matematiske sammenhæng er ren og skær tilfælde. Husk på at foldning og korrelation er to meget forskellige procedurer.

Foldning er sammenhængen mellem et systems input signal, output signal og impulsrespons.

Korrelation er en måde at detektere et kendt signal i et støjfyldt signal.

Hvis vi sender hvid støj ind som input,  $x(n) = \text{hvid støj}$ , bliver kryds-korrelationen lig:

$r(n) = x(n) * y(-n)$	Outputtet er lig filter gange input: $y(n) = h(n) * x(n)$
$r(n) = x(n) * h(-n) * x(-n)$	Input-signalet foldet med sig selv er lig delta-funktionen: $x(n) * x(n) = \delta(n)$
$r(n) = h(-n) * \delta(n) \cdot \sigma^2$	Et filter foldet med delta-funktionen er lig 1: $h(n) * \delta(n) = 1$
$r(n) = h(-n) \cdot \sigma^2$	$r(n)$ er et udtryk for, hvor meget et signal ligner et andet signal - tidsforskuet med $n$ samples

*Auto-korrelation* er et udtryk for hvor meget et signal ligner sig selv, når vi folder det med sig selv. Ved lag nummer 0, vil korrelationen være størst, fordi det er her signalet overlapper sig selv under foldningen.

#### **DISPOSITION 11 - CASE 1 FSK**

#### **DISPOSITION 12 - CASE 2 Audio Notch Filter**

#### **DISPOSITION 13 - CASE 3 Vejecelle**

Når vi sammenligner de to histogrammer, er opløsningen forskellig

#### **DISPOSITION 14 - CASE 4 Sonar**

Under x-korrelationen vender vi arrayet om og folder det med det oprindelige. Dette resulterer i den største peak når de to signaler overlapper hinanden fuldstændig.