

---

## Tor Halmrast

Komponist, Sjøfing. Akustikk

1.amanuensis II, Univ.Oslo, inst. for Musikkvitenskap

privat: Spångberg vn. 28a, 0853 OSLO

(+47) 22230635

mobile phone: (+47)95191675

e-mail private: [torhalm@online.no](mailto:torhalm@online.no)

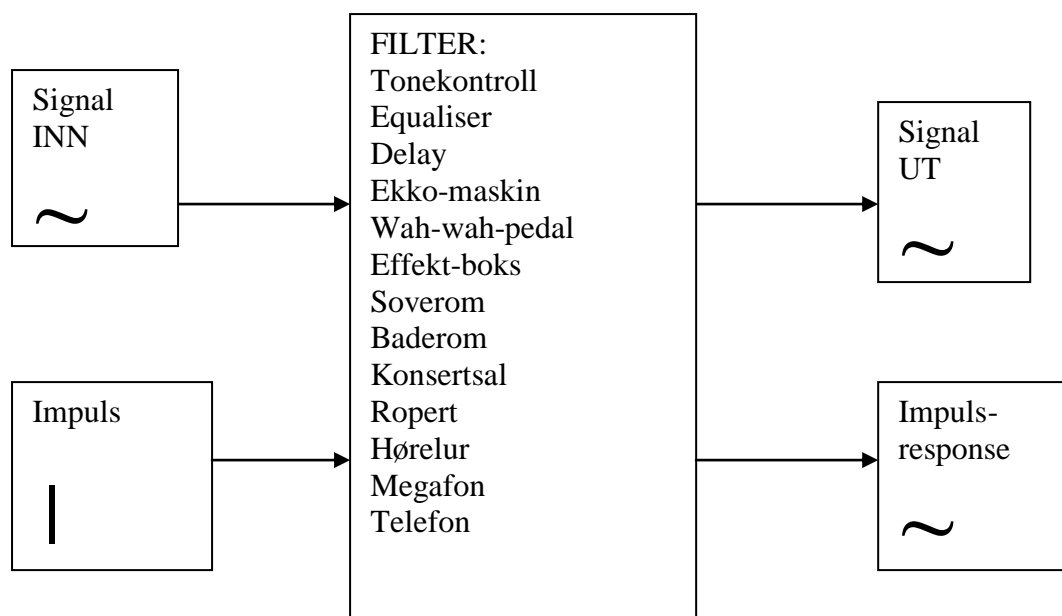
<http://tor.halmrast.no/>

---

## LYDANALYSE h 2013

### FILTRE

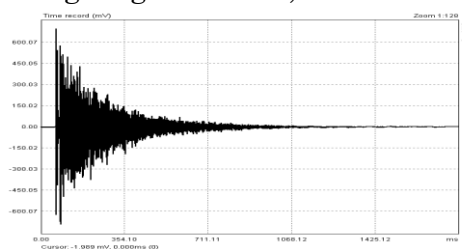
ALT VI KAN SENDE SIGNAL IGJENNOM KAN FAKTISK KALLES ET "FILTER"



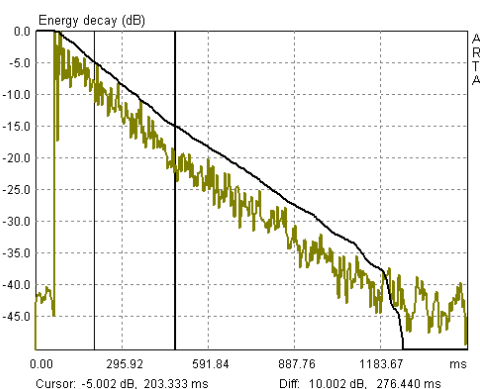
Alle "filtre" kan karakteriseres ved sin **Impulsrespons**.

Det er utgangssignalet når inngangssignalet er en kort puls, ideelt en såkalt **"Dirac-puls"**, som er kun ett sample lang, men som allikevel inneholder alle frekvenser (noe som kanskje kan virke selvmotsigende). Dersom man ikke har en slik ideell "enhets-puls" tilgjengelig, er det omtrent det samme man får dersom man skyter et **pistolskudd** eller klapper meget sterkt og kort for å sjekke akustikken i en sal, tapper fingeren på en mikrofon, eller når man gir et **svingesystem** (f.eks en trampoline eller en vaskemaskin på vibrasjonsisolatorer) et meget kort slag/dytt. I slike tilfelle, med en ikke-ideell puls som kilde, vil man selvfølgelig ikke få informasjon for annet enn det frekvensområdet signalet har energi, f.eks vil et håndklapp neppe eksitere særlig bass. Dessuten vil en slik "nesten-impuls" være for lang i forhold til "ett sampel", slik at det vil utjevne små differenser, og ikke være særlig god for høy diskant. Men for å øke forståelsen av hva en impulsrespons er, kan vi melde oss inn i en pistolklubb.

En **impulsrespons** viser trykksvingningene (eller egentlig spennings-svingningene i Volt, som kommer ut av mikrofonen) over tid:



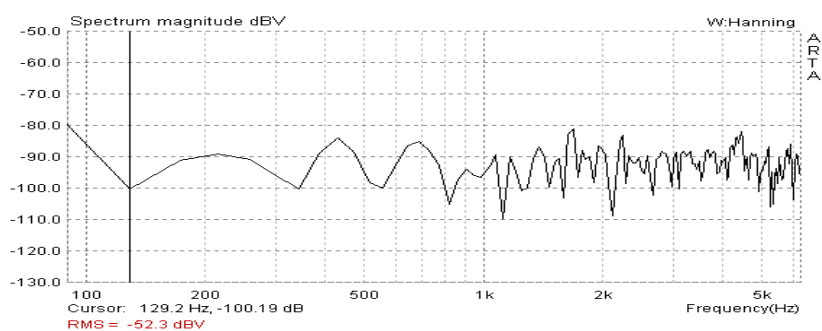
Impuls-responsen blir ofte **kvadrert**, noe som tilsvarer lydenergien: (ETC- Energy-Time-Curve)



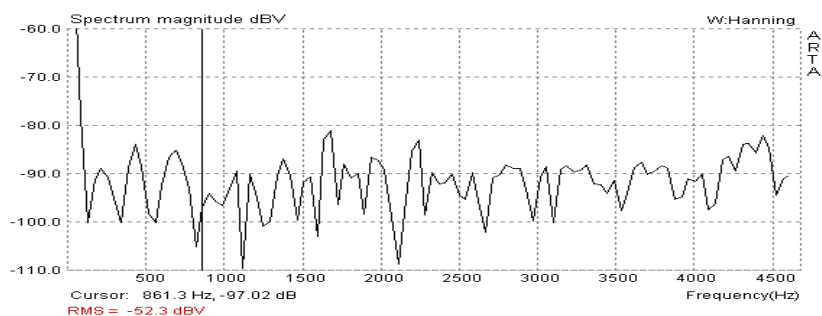
Ut i fra en slik Impuls-Respons kan vi kalkulere f.eks.

## FrekvensRespons

(vha. FFT, Fast Fourier Transform). Her med logaritmisk frekvensakse:

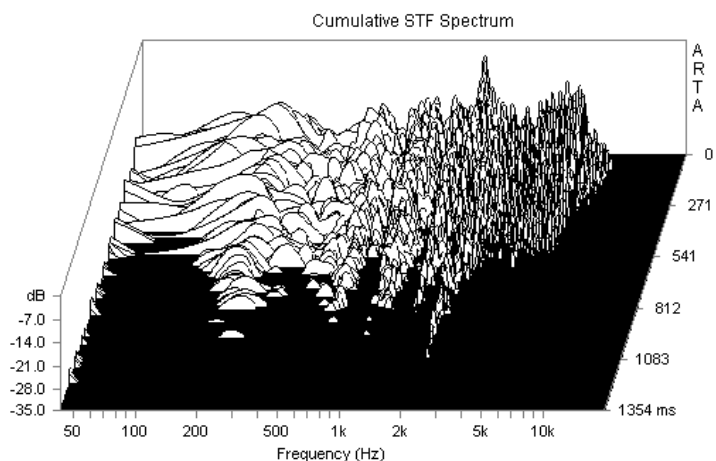


Eller med lineær frekvensakse:

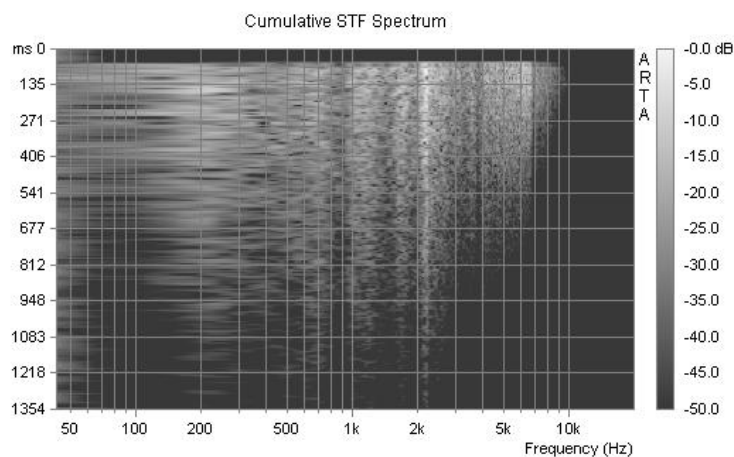


Dersom "filteret" er f.eks. et rom, kan vi beregne ALLE interessant rom-akustiske parametere ut fra Impulsresponsen, f.eks. Etterklangstid

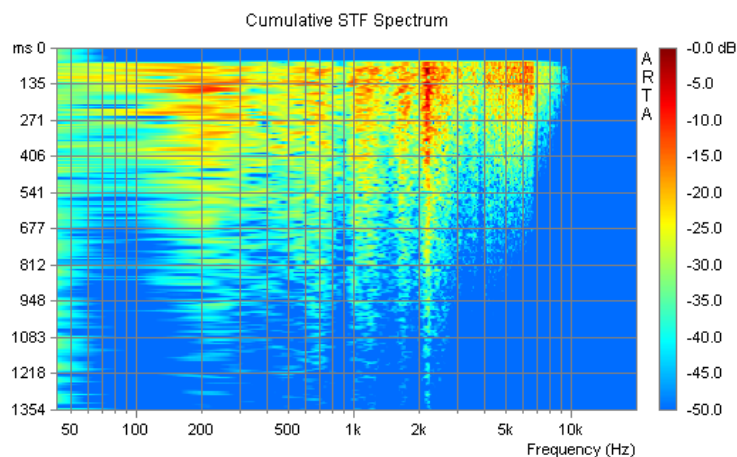
Responsen kan også vises i såkalt **Waterfall**:



eller i **spectrogram/sonogram**:



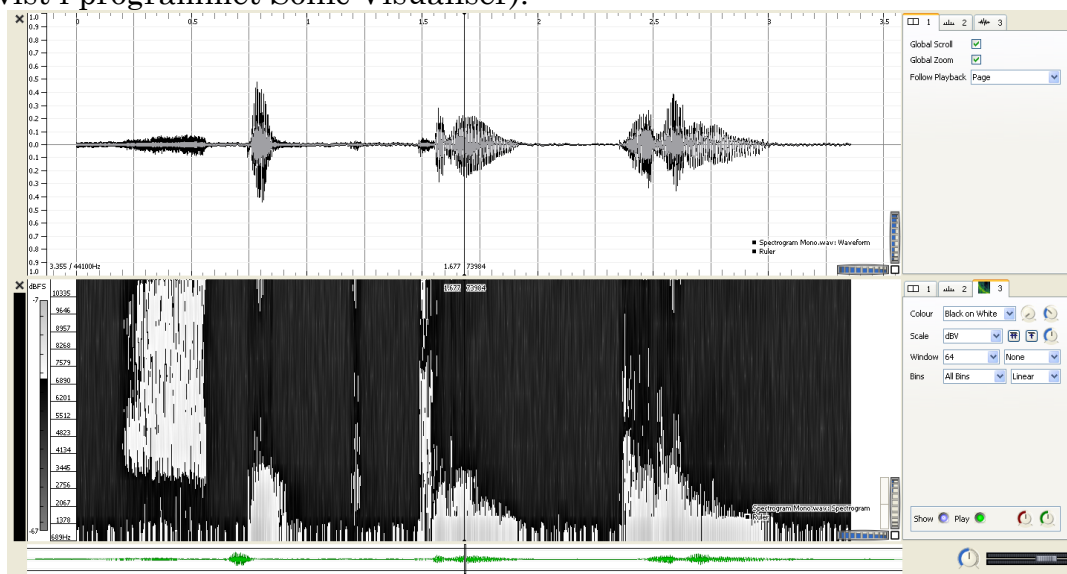
Slike ser selvfølgelig finere ut i farger:



## 11.6 Når skjer det? Hva er det? Presisjon i Tid og Frekvens

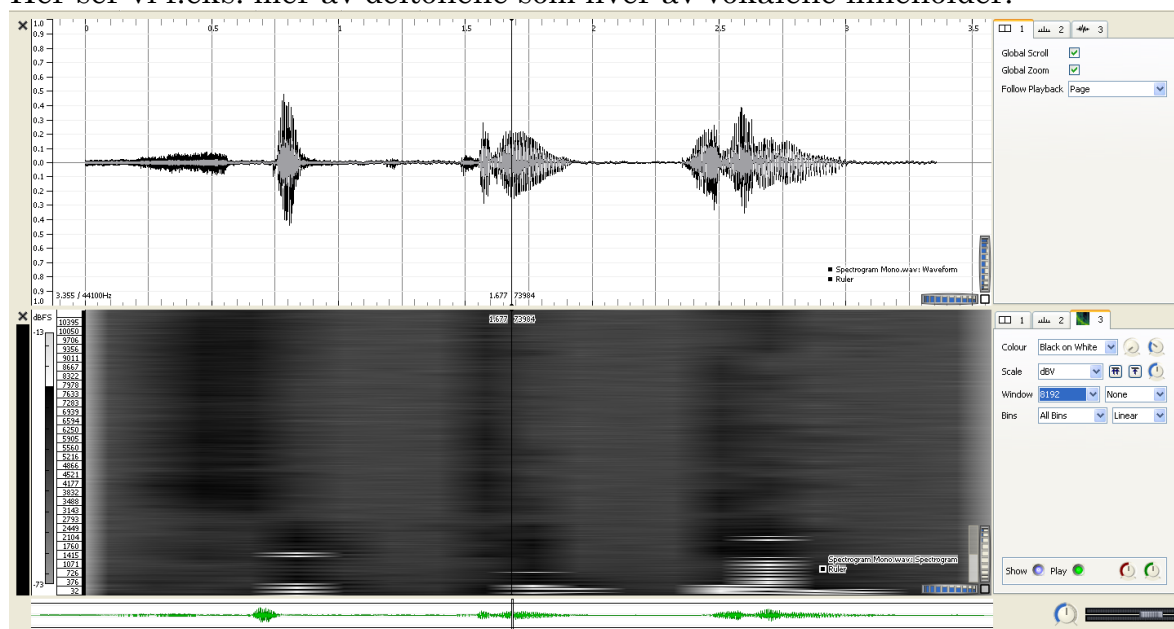
Fourier-analyse gjelder egentlig kun for uendelig lange signaler. Dessuten gir vanlig Fourier-analyse det samme frekvens-resultat om det er flere toner samtidig eller de kommer etter hverandre. For å finne ut noe om når signalene kommer, kan vi prøve å ta Fourier-analyse av kortere tidsintervaller, såkalte Vinduer (eng: Windows). Så flytter vi dette vindu litt i tid, og tar en ny analyse. Dette kalles Kort-tids-Fourier-analyse (**Feil! Fant ikke referansekilden., SFT**). Vi må da benytte korte Vinduer for å få nøyaktighet i tid, og lange Vinduer for å få nøyaktighet i Frekvens. Vi kan IKKE få god nøyaktighet i både tid og frekvens i samme analyse. Dette kan sees på som en variant av Heisenberg Usikkerhetslov.

En FFT-analyse av ordet ”*Spectrogram*” har vi allerede vist, (se [spektrogrammer](#)). Dersom vi ønsker å finne mest mulig er nøyaktig tidspunkt for hver lydhendelse, må vi sette et kort vindu:  
(Her vist i programmet Sonic Visualiser).



Dersom vi ønsker bedre nøyaktighet i frekvens, må vi sette et lengre vindu.

Her ser vi f.eks. mer av deltonene som hver av vokalene inneholder:



Legg også merke til at ved slik kort vindu, kan man heller ikke analysere så langt ned i frekvens som med et langt vindu.

Med langt vindu, og mye overlapp mellom vinduene, samt en logaritmisk frekvensakse, kan vi få ganske god informasjon om hvilke ”musikalske toner” vokalene inneholder, og at det er en liten glissando på slutten av ”-am” i den siste delen av ordet: ”...gram”. Ved logaritmisk frekvensskala viser Sonic Visualiser dessuten et meget lite ”piano-klaviatur” langs frekvensaksen.

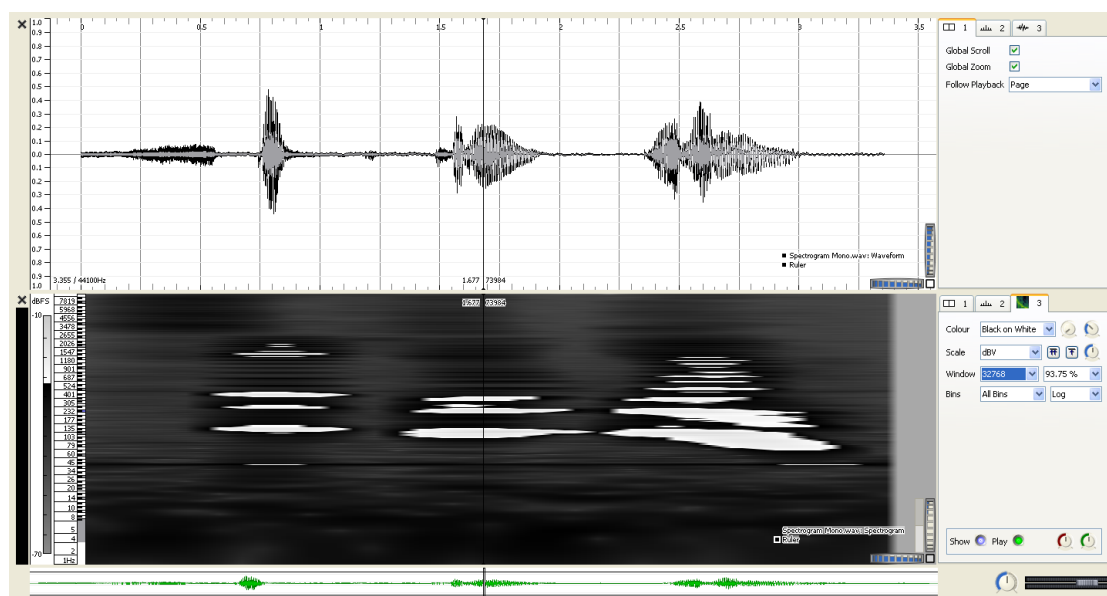
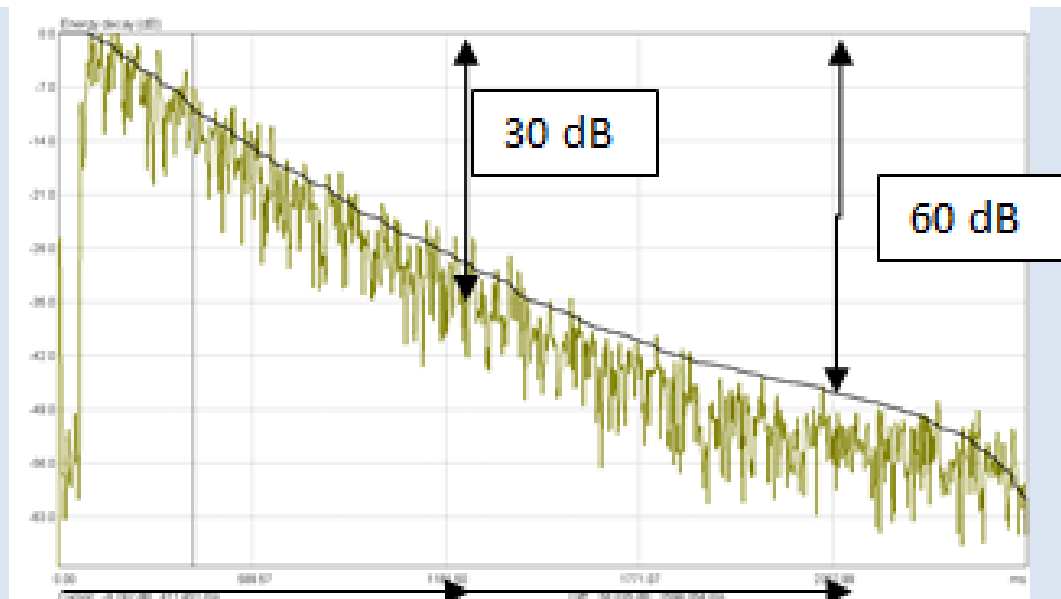


fig: TH

## ETTERKLANGSTID

Tiden det tar før lyden har fallt 60 dB etter at den er skrudd av:



$$T = T_{60} = RT_{60} = \text{ca } 2,4 \text{ s}$$

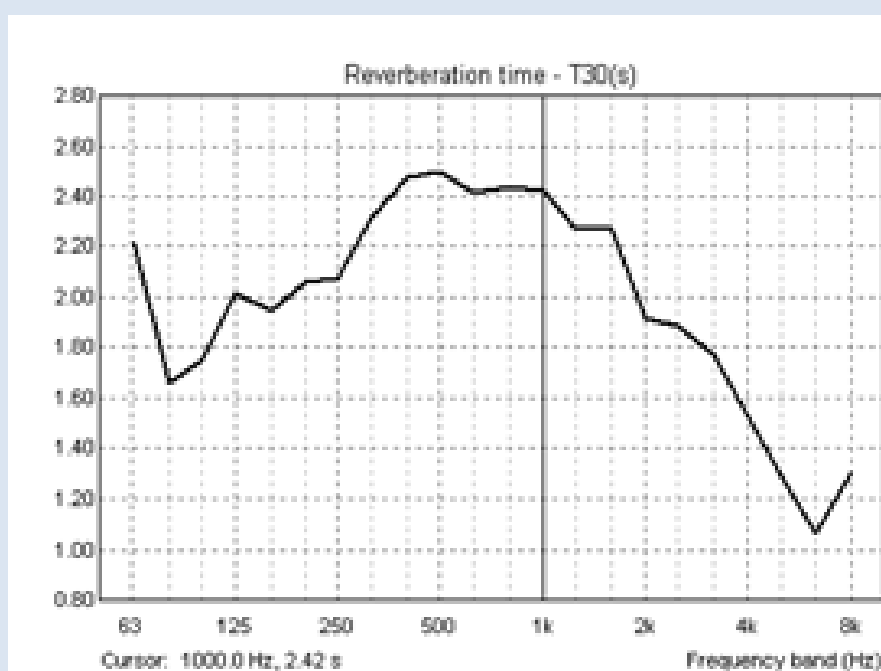


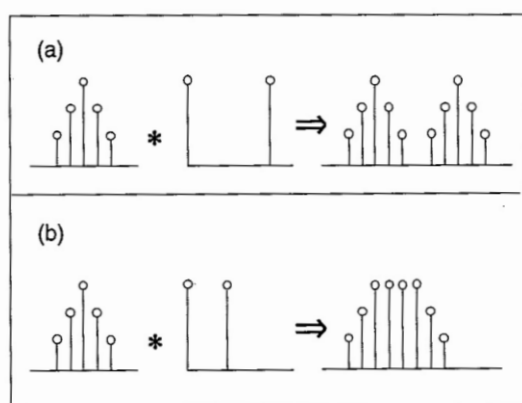
fig: TH

## Konvolusjon (Convolution)

Dette er en meget interessant del av lydbehandling, og den skjer faktisk hver dag, hjemme eller i konsertsalen. I begge tilfelle har vi et, i utgangspunktet, "tørt" signal, dvs. uten klang. Dette kan være min hosting, eller instrumentene på scenen i Konserthuset. Dette lydsignal sendes så ut i et rom. Hvis dette rom ikke er helt "dødt" vil vi få refleksjoner fra vegger etc. Matematisk beskrives dette med "convolution". I romakustikk "konvolverer" vi det "tørre" opptaket av musikken (eller min hosting) med Impulsresponsens av rommet. Impulsresponsen inneholder da alle refleksjoner lyden gjennomgår i rommet.

Tilsvarende kan man selvfølgelig "konvolvere" en "tørr" gitar-lyd med Impulsresponsen fra en stor Marshall, og ut-signalet skal da (teoretisk) tilsvare som om du hadde råd til en Marshall.

Convolution for et digitalt signal kan vises slik:



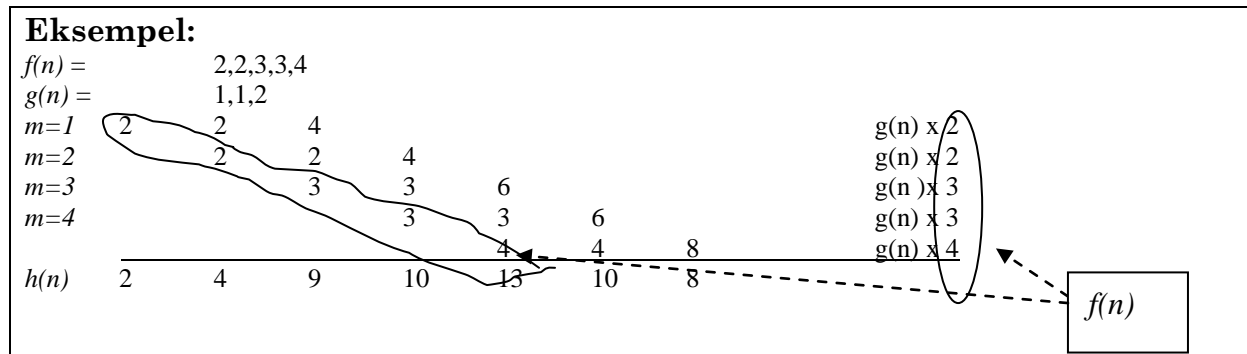
Impulsresponsen for filteret/rommet/effektboksen er her en enkel tidsforsinkelse. I fig a) er forholdet ganske enkelt. Her er signalet så kort, og tidsforsinkelsen så lang, at hele signalet blir ferdig før det kommer tidsforsinket. Dette blir da et distinkt ekko. I fig b) er tidsforsinkelsen såpass kort at signalet ikke er verdig før det kommer tidsforsinket. Da blir signalet og den tidsforsinkede variant "smurt sammen", konvulert. Hva nå hvis "filteret"/rommet/effektboksen var litt mer kompleks enn bar denne "ekko-maskinen"?

### Definisjon av "convolution"

Konvulering ("convolution") betyr at vi "erstatte" hvert enkelt sample av den ene funksjonen, med alle samples av den andre funksjonen, og legger til overlappende verdier.

Konvulering angis ofte med et slags alternativt multiplikasjonstegn,  $\otimes$ , eller  $*$ . Dette siste tegnet benyttes (dessverre) også som vanlig multiplikasjon f.eks. i Excel, men konvulsjon er altså mer komplisert, og kan matematisk skrives slik:

$$h(n) = f(n) * g(n) = \sum_{m=0}^n f(m)g(n-m)$$



For mer morsomme effekter, kan man ”gange”/konvulere en lyd med en annen lyd, eller kanskje med seg selv? (evt. tidsforsinket).

Ved konvulering **”ganger man i frekvensplanet”**, dvs at dersom de to signalene, på det aktuelle tidspunkt, inneholder samme frekvenser, vil disse komme fram i utgangssignalet.

## Konvulering i tidsdomenet tilsvarer multiplikasjon i frekvens-domenet.

Vi multipliserer altså frekvensresponsene med hverandre.

Når vi skal beregne utgangssignalet fra et filter, konvulerer vi altså inngangssignalet med filterets Transfer Function/Impulse Response

Hvis vi kaller de to signalene vi konvulerer med hverandre henholdsvis  $f$  (med lengde  $L(f)$ ) og  $g$  (med lengde  $L(g)$ ), blir lengden av utgangssignalet  $h$ :  $L(h) = L(f) + L(g) - 1$ .

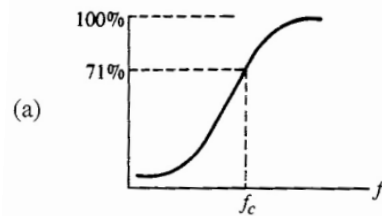
## KONVULERING i PD



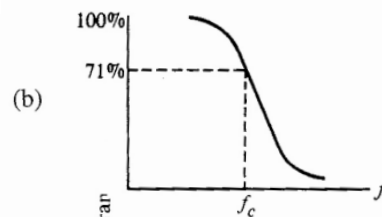
## De enkleste typer filtre

### (både analoge og digitale)

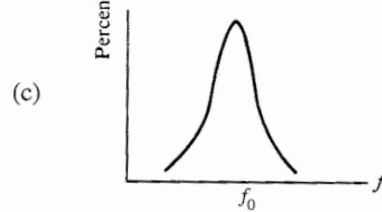
a) Høy-pass ("High pass")



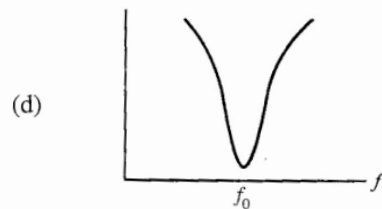
b) Lav-pass ("Low pass")



c) Bånd-pass ("Band pass")

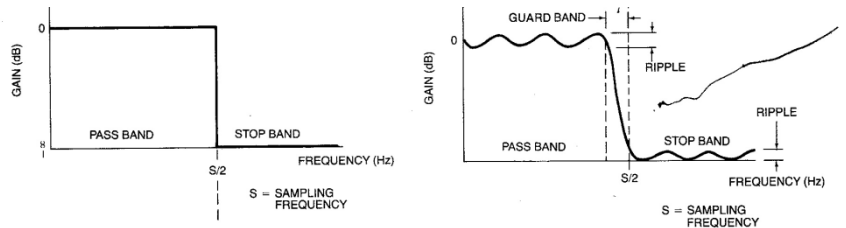


d) Band-reject ("Band reject")

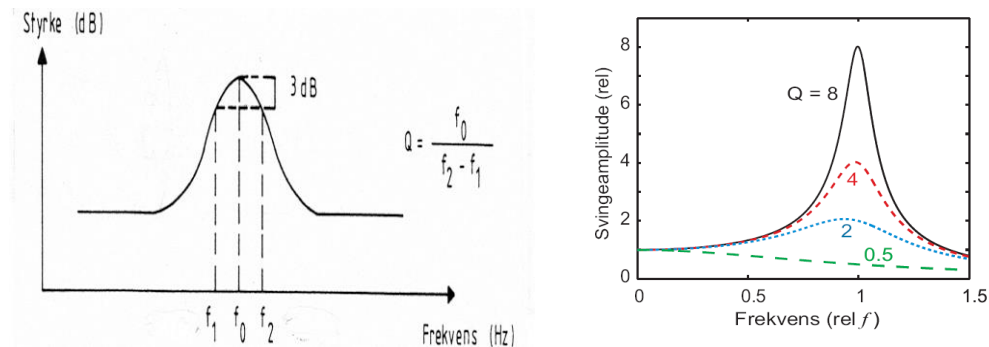


Her angitt med sine resonansfrekvenser  $f_0$ .

Man skulle ønske at filtre kapper rett av („instantaneous cutoff“), slik at absolutt ingen frekvenser under en viss grense kommer med og alle over ikke kommer med, men i praksis får man noe „søling“ også f.eks over frekvensen et lavpassfilter skal skjære ved:



## Q-faktor, Båndbredde



Båndbredden ( $f_2 - f_1$ ) til et filter defineres med den såkalte **Q-verdien**.

Forholdet mellom midtfrekvensen  $f_0$  og frekvensbåndbredden på karakteristikken 3 dB under midtfrekvensen.

Jo lavere Q-verdien er, desto større blir frekvensbåndbredden.

En **Impuls Respons** kan angis som en **Transfer Function** for et digitalt filter. Her kan man komme bort i masse underlige ting med negative frekvenser etc, som vi ikke skal ta opp her, men konsentrere oss om at ut fra Impulsresponsen fra et filter, kan vi finne dets frekvensrespons

For å bygge et digitalt filter lager vi en operator kalt

## Unit Delay Operator $Z^{-1}$

Når man anvender denne operator på en sekvens av digitale verdier, får man den forrige verdien i sekvensen. Det betyr altså i praksis en tidsforsinkelse ("delay") på ett sampel. Hvis man anvender denne operatoren på en inngangsverdi (f.eks.  $x_n$ ), får man altså den forrige verdien ( $x_{n-1}$ ):

$$Z^{-1} x_n = x_{n-1}$$

Anta at vi har en sekvens:

$$x_0 = 5$$

$$x_1 = -2$$

$$x_2 = 0$$

$$x_3 = 7$$

$$x_4 = 10$$

Da blir:

$$Z^{-1} x_1 = x_0 = 5$$

$$Z^{-1} x_2 = x_1 = -2$$

$$Z^{-1} x_3 = x_2 = 0, \text{ osv.}$$

PS!  $Z^{-1} x_0$  skulle da bli  $x_{-1}$  som er ukjent, men som oftest settes lik null.

På tilsvarende måte: Når vi benytter  $Z^{-1}$  operatoren på en utgangsverdi, får vi den forrige utgangsverdien:

$$Z^{-1} y_n = y_{n-1}$$

Dersom vi benytter operatoren  $Z^{-1}$  to ganger, får vi en forsinkelse ("delay") på to samples:

$$Z^{-1} (Z^{-1} x_n) = Z^{-1} x_{n-1} = x_{n-2}$$

## Ikke-rekursive Filtre = FIR

(Finite Impulse Response) filter:

Utgangen beregnes kun fra den nåværende og de tidligere input-verdier:

$(x_n, x_{n-1}, x_{n-2}, \dots)$ .

## Rekursive Filtre= IIR

(Infinite Impulse Response) filter

tar i tillegg de tidligere utgangsverdiene med i betraktning, altså ikke bare

$(x_n, x_{n-1}, x_{n-2}, \dots)$ , men også  $y_{n-1}, y_{n-2}, \dots$

Rekursive betyr ”å løpe bakover”.

Ut fra denne forklaring skulle man tro at rekursive filtre krever flest kalkulasjoner, siden vi jo må ta hensyn også til foregående verdier. I praksis er dog det motsatte vanligvis tilfelle. For å oppnå en gitt frekvensrespons-karakteristikk krever et rekursivt filter som regel en lavere filter-orden, og derfor færre ledd som skal beregnes, enn for et Ikke-rekursivt filter.

En Impulsrespons får man jo ved å sende en ”enhets impuls” til inngangen av filteret. (En enhets-impuls” er en meget enkel sekvens som inneholder kun en verdi på 1 ved  $t=0$ , fulgt av bare nuller på alle resterende plasser. Et FIR-filter har Impulsrespons som har endelig lengde. Et rekursivt (IIR) filter har (teoretisk) en uendelig lang impulsrespons, fordi man stadig tar med den forrige verdien i beregningen. Terminologien ”Infinite” er dog ikke helt riktig, da de fleste IIR-filtre reduserer til praktisk talt null etter en endelig tid.

## MÅTER Å ANGI FILTRE PÅ:

### 1) Filter-PARAMETRE

### 2) ZERO/POLE

# Filter orden

Filtre settes i grupper/"orden" etter hvor mange av de foregående inngangssignalene (lagret i prosessorens minne) som benyttes i kalkulasjonen av det nåværende utgangssignal.

## NULL' TE ORDEN

**Eksempel (1):**  $y_n = x_n$

Dette er et "null'te ordens filter", siden det nåværende utgangssignal  $y_n$  kun er avhengig av det nåværende inngangssignal  $x_n$  og ikke av noen tidligere innkomne inngangssignal.

**Eksempel (2):**  $y_n = Kx_n$

Dette filter er også av "null'te orden", av samme grunn som over.

## FØRSTE ORDEN

**Eksempel (3):**  $y_n = x_{n-1}$

Dette er et "første ordens filter", siden ett av de forgående inngangssignal ( $x_{n-1}$ ) er nødvendig for å kalkulere  $y_n$ . (Merk at dette filter er av første orden, fordi det kun benytter ett inngangssignal, selv om det nåværende inngangssignal ikke benyttes).

**Eksempel (4):**  $y_n = x_n - x_{n-1}$

Dette er igjen et "første ordens filter", siden ett tidligere inngangssignal er nødvendig for å gi det nåværende utgangssignal.

**Eksempel (5):**  $y_n = (x_n + x_{n-1}) / 2$

Dette er også et "første ordens filter", siden det benytter kun et tidligere inngangssignal.

## ANDRE ORDEN

**Eksempel (6):**  $y_n = (x_n + x_{n-1} + x_{n-2}) / 3$

For å beregne det nåværende utgangssignal  $y_n$  trengs to tidligere inngangssignal ( $x_{n-1}$  and  $x_{n-2}$ ). Dette er derfor et "annen ordens filter".

**Eksempel (7):**  $y_n = (x_n - x_{n-2}) / 2$

Filterordenen er igjen 2, siden prosessoren må lagre to tidligere inngangssignal for å beregne det nåværende utgangssignal. Dette påvirkes ikke av at  $x_{n-1}$  leddet ikke er med.

Filterets orden kan være ethvert positivt heltall. Et "null'te ordens filter" (slik som i Eksempel 1 og 2, mulig, men ganske trivielle, siden de ikke egentlig gir noen filtrering i den vanlige betydning, en evt. en amplitude- endring (konstanten K)

I de følgende Excel-ark er et vilkårlig "signal" angitt, og man kan sette inn filterkoeffisientene  $A_0, A_1$ , for et 1.ordens filter, og dessuten  $A_2$  for 2.orden.

UJO, Inst Musikkvitenskap  
Tor Halmrast ( ut fra Kristians utmerkede oppgave)  
**DIGITALE FILTER**  
apr.06

### FIR-filter

No feedback  
Stable, symmetrical, phase neutral, avoid overflow.  
May require more processing and memory than IIR-filters  
FIR easier to design

Filter Order	ENKELT	MED KOEFFISIENTER
0) Zero order	$y_n = x_n$ $y_n = K \cdot x_n$	
1) First order	$y_n = x_{n-1}$ $y_n = K \cdot x_{n-1}$ $y_n = (x_n + x_{n-1})/2$	$y_n = A_0 \cdot x_n + A_1 \cdot x_{n-1}$
2) Second order	$y_n = x_n + x_{n-1} + x_{n-2}$ $y_n = (x_n + x_{n-1} + x_{n-2})/2$	$y_n = (A_0 \cdot x_n + A_1 \cdot x_{n-1} + A_2 \cdot x_{n-2})$

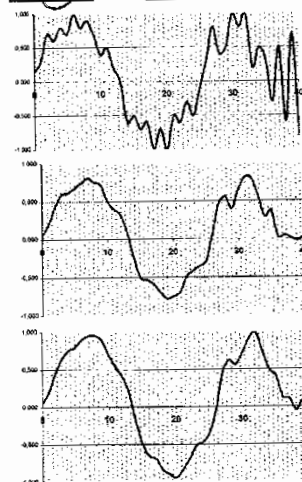
(uses the previous input, even though the current input is not used)  
(1. order filter, since one previous value is required)

(To compute the current output,  $y_n$ , two previous inputs are needed,  
(this is therefore a second-order filter)

**FILTER KOEFFISIENTER**  
(samme fortløp for A'ene gir Lavpass, forskjellig fortløp gir Høypass)  
Eks:  $y_n = 0.5x_n + 0.5x_{n-1}$  Tar midt middelet, glatter  
 $y_n = 0.5x_n - 0.5x_{n-1}$  Tar med forskjellen mellom sample, altså high-pass

For FIR-IIR:  $A_0, A_1, A_2$   
For IIR:  $B_1, B_2$

Sample nr	Innverdi	Filter order1 Utverdi $A_0, A_1$	Filter order2 Utverdi $A_0, A_1, A_2$
0	0.150	0.060	0.060
1	0.300	0.195	0.195
2	0.700	0.430	0.460
3	0.600	0.590	0.650
4	0.800	0.620	0.760
5	0.700	0.680	0.800
6	0.990	0.745	0.968
7	0.800	0.815	0.855
8	0.500	0.760	0.855
9	0.700	0.730	0.890
10	0.400	0.510	0.690
11	0.500	0.460	0.540
12	0.200	0.330	0.410
13	0.000	0.100	0.200
14	-0.600	-0.240	-0.200
15	-0.500	-0.520	-0.500
16	-0.700	-0.530	-0.650
17	-0.800	-0.590	-0.680
18	-0.990	-0.698	-0.636
19	-0.700	-0.775	-0.695
20	-0.990	-0.746	-0.844
21	-0.500	-0.695	-0.835
22	-0.600	-0.490	-0.888
23	-0.300	-0.420	-0.520
24	-0.500	-0.350	-0.470
25	-0.100	-0.290	-0.350
26	0.300	0.070	-0.030
27	0.800	0.470	0.450
28	0.400	0.550	0.620
29	0.500	0.400	0.590
30	0.890	0.846	0.726
31	0.800	0.815	0.815
32	0.990	0.796	0.904
33	0.200	0.575	0.735
34	0.500	0.300	0.468
35	0.300	0.370	0.410
36	-0.300	0.030	0.130
37	-0.500	0.090	0.119
38	-0.900	0.010	-0.050
39	0.700	-0.020	0.080
40	-0.800	0.030	-0.080



FIR, 1.ordens filter:

$$y_n = A_0 x_n + A_1 x_{n-1}$$

Både  $A_0$  og  $A_1$  positive her, (f.eks + 0,5), gir:

Midling av to og to sampel-verdier, derfor:

### Lavpassfilter

Under tilsvarende for 2.ordens filter,

og for IIR-filtre:

### IIR filter

$$y_n = A_0 x_n + A_1 x_{n-1} + \dots + A_N x_{n-N} + B_1 y_{n-1} + \dots + B_M y_{n-M}$$

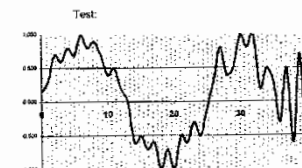
The output sample depends on the previous output samples

IIR filters are the counterpart to Analog Filters. They use FEEDBACK, which normally requires less computing times than FIR.

Due to feedback, high order IIR may have instabilities, and OVERFLOW

Additionally: phase shift

Sample nr	Innverdi	UI: $A_0, A_1, B_1$	UI: $A_0, A_1, A_2, B_1, B_2$
0	0.150	0.060	0.060
1	0.300	0.225	0.150
2	0.700	0.643	0.375
3	0.600	0.861	0.726
4	0.800	1.051	1.028
5	0.700	1.206	1.167
6	0.990	1.346	1.207
7	0.800	1.489	1.236
8	0.900	1.535	1.391
9	0.700	1.482	1.393
10	0.400	1.251	1.243
11	0.500	1.026	0.981
12	0.200	0.843	0.709
13	0.000	0.521	0.504
14	-0.600	0.021	0.167
15	-0.500	-0.490	-0.354
16	-0.700	-0.775	-0.815
17	-0.600	-0.877	-1.009
18	-0.990	-1.185	-1.059
19	-0.700	-1.387	-1.155
20	-0.990	-1.430	-1.309
21	-0.500	-1.410	-1.351
22	-0.600	-1.195	-1.204
23	-0.300	-1.017	-0.931
24	-0.500	-0.859	-0.716
25	-0.100	-0.719	-0.619
26	0.300	-0.290	-0.422
27	0.800	0.325	0.079
28	0.400	0.723	0.702
29	0.500	0.761	0.978
30	0.890	1.027	0.929
31	0.800	1.328	1.020
32	0.990	1.490	1.284
33	0.200	1.305	1.389
34	0.500	0.953	1.081
35	0.300	0.846	0.861
36	-0.300	0.453	0.394
37	-0.500	0.277	0.204
38	-0.900	0.148	0.102
39	0.700	0.054	0.045
40	-0.800	0.057	0.021



## FIR-filter

No feedback

Stable, symmetrical, phase neutral, avoid overflow.  
 May require more processing and memory than IIR-filters  
 FIR easier to design

Filter Order	ENKELT	MED KOEFFISIENTER
0) Zero order	$y_n = x_n$ $y_n = K \cdot x_n$	
1) First order	$y_n = x_{n-1}$ $y_n = x_n + x_{n-1}$ $y_n = (x_n + x_{n-1})/2$	$y_n = A_0 \cdot x_n + A_1 \cdot x_{n-1}$
2) Second order	$y_n = (x_n + x_{n-2})$ $y_n = (x_n - x_{n-2})/2$	$y_n = (A_0 \cdot x_n + A_1 \cdot x_{n-1} + A_2 \cdot x_{n-2})$

(uses the previous input, even though the current input is not used)  
 (1. order filter, since one previous value is required)

(To compute the current output,  $y_n$ , two previous inputs are needed,  
 (this is therefore a second-order filter)

### FILTER-KOEFFISIENTER

(samme fortegn for A'ene, gir Lavpass, forskjellig fortegn gir Høypass)  
 Eks:  $y_n = 0.5x_n + 0.5x_{n-1}$  Tar med middelet, glatter  
 $y_n = 0.5x_n - 0.5x_{n-1}$  Tar med forskjellen mellom samplene, altså high-pass

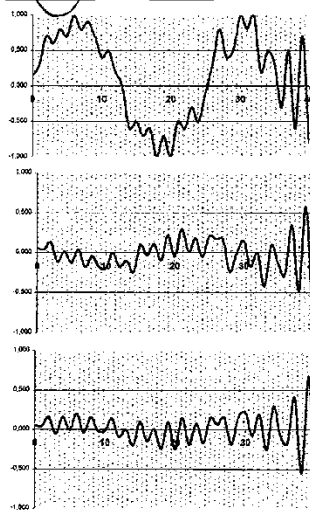
For FIR+IIR

$A_0$	$A_1$	$A_2$
0.4	0.6	0.2

For IIR

$B_1$	$B_2$
0.5	-0.5

Sample nr	$x_n$	Filter order1 Utverdier $A_0$ og $A_1$	Filter order2 Utverdier $A_0$ , $A_1$ og $A_2$
0	0.150	0.000	0.000
1	0.300	0.045	0.045
2	0.700	0.130	0.160
3	0.600	-0.110	-0.050
4	0.800	0.020	0.160
5	0.700	-0.120	0.000
6	0.990	0.000	0.298
7	0.800	-0.175	-0.036
8	0.900	-0.040	0.158
9	0.700	-0.170	-0.010
10	0.400	-0.190	-0.010
11	0.500	0.000	0.140
12	0.200	-0.170	-0.090
13	0.000	-0.100	0.000
14	-0.600	-0.240	-0.200
15	-0.500	0.100	0.100
16	-0.700	-0.030	-0.150
17	-0.600	0.110	0.010
18	-0.990	-0.090	-0.236
19	-0.700	0.215	0.085
20	-0.980	-0.040	-0.244
21	-0.500	0.235	0.155
22	-0.600	-0.010	-0.188
23	-0.300	0.180	0.080
24	-0.500	-0.060	-0.170
25	-0.100	0.210	0.150
26	0.300	0.170	0.070
27	0.800	0.170	0.150
28	0.400	-0.240	-0.180
29	0.630	0.002	0.180
30	0.990	0.146	0.228
31	0.800	-0.175	-0.075
32	0.990	-0.004	0.194
33	0.200	-0.415	-0.295
34	0.500	0.100	0.298
35	0.300	-0.130	-0.090
36	-0.950	-0.270	-0.170
37	0.990	0.350	0.410
38	-0.800	-0.490	-0.590
39	0.700	0.580	0.580
40	-0.800	-0.670	-0.790



$A_0$  og  $A_1$  forskjellige  
 fortegn, f.eks  $A_1 = -0.5$  her,  
 gir  
 Differensen mellom to og to  
 sampels,  
 Altså:

**Høypassfilter**

Excel etter Kristian Nymoen

## IIR filter

$$y_n = A_0 x_n + A_1 x_{n-1} + \dots + A_N x_{n-N} + B_1 y_{n-1} + \dots + B_M y_{n-M}$$

The output sample depends on the previous output samples

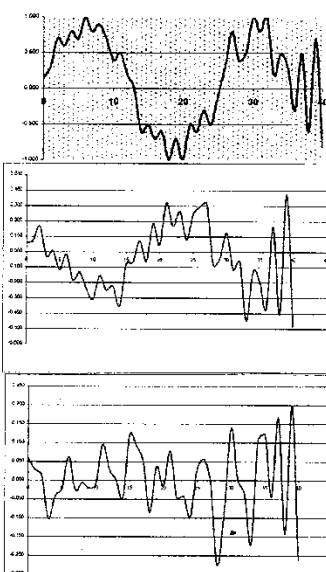
IIR filters are the counterpart to Analog Filters. They use FEEDBACK, which normally requires less computing times than FIR.

Due to feedback, high order IIR may have problems with instability, and OVERFLOW

Additionally: phase shift

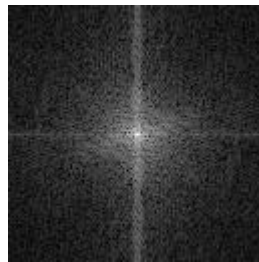
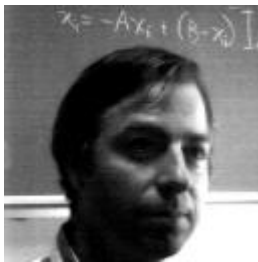
Sample nr	Innverdier	Utverdier $A_0, A_1, B_1, B_2$	Utverdier $A_0, A_1, B_1, B_2$
0	0.150	0.060	0.060
1	0.300	0.075	0.030
2	0.700	0.188	0.015
3	0.600	-0.026	-0.102
4	0.800	0.007	-0.042
5	0.700	-0.117	-0.020
6	0.990	-0.012	0.084
7	0.800	-0.181	-0.024
8	0.990	-0.131	-0.006
9	0.700	-0.235	-0.020
10	0.400	-0.308	-0.012
11	0.500	-0.154	0.096
12	0.200	-0.247	0.032
13	0.000	-0.223	0.002
14	-0.600	-0.352	-0.044
15	-0.500	-0.076	0.122
16	-0.700	-0.068	0.094
17	-0.600	0.076	0.050
18	-0.990	-0.059	-0.083
19	-0.700	0.186	0.037
20	-0.990	0.047	-0.014
21	-0.500	0.319	0.078
22	-0.600	0.169	-0.046
23	-0.300	0.265	-0.041
24	-0.500	0.082	-0.097
25	-0.100	0.251	0.031
26	0.300	0.296	0.057
27	0.800	0.318	-0.008
28	0.400	-0.081	-0.224
29	0.630	-0.041	-0.084
30	0.990	0.126	0.141
31	0.800	-0.112	0.004
32	0.990	-0.060	-0.036
33	0.200	-0.445	-0.168
34	0.500	-0.123	0.107
35	0.300	-0.191	0.123
36	-0.900	-0.366	-0.043
37	0.990	-0.167	0.167
38	-0.800	-0.495	-0.141
39	0.700	0.377	0.200
40	-0.800	-0.482	-0.210

Test:

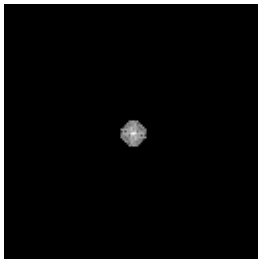


## Filtre i Billedbehandling

Filtrering kan også benyttes i billedbehandling. Her er noen eksempler:



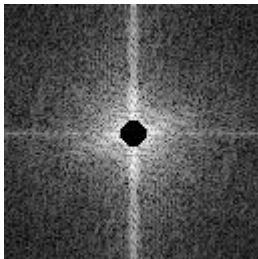
**Low-Pass Filtered**



**Inverse  
Transformed**



**High-Pass Filtered    Inverse Transformed**

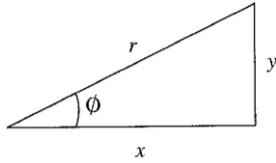




# KOMPLEKSE TALL/VEKTORER

fra Kompendium Lydteori1:

## LITT TRIGONOMETRI



$$y = A \sin \phi = A \sin 360 \frac{t}{T}$$

der 360 er antall grader i en full sirkel, T er hele perioden/sirkelbegelsen, og t er tiden.

$$\sin \phi = \frac{y}{r} \quad \text{eksempel: } \sin 0 = 0, \sin 90 = 1, \sin 45 = \frac{1}{\sqrt{2}}, \sin 135 = \sin 45$$

$$\cos \phi = \frac{x}{r} \quad \text{eksempel: } \cos 0 = 1, \cos 90 = 0, \cos 45 = \frac{1}{\sqrt{2}}, \cos 135 = -\cos 45$$

$$\tan \phi = \frac{y}{x} \quad \text{eksempel: } \tan 0 = 0, \tan 90 = \infty, \tan 45 = 1, \tan 135 = -\tan 45$$

Ofte antas  $r=1$  (unit) og man setter i stedet en Amplitude som fast tillegg for alle verdier

$$x^2 + y^2 = 1 \quad (\text{Pythagoras})$$

$$\cos^2 \phi + \sin^2 \phi = 1$$

## ALTERNATIV MATEMATISK BESKRIVELSE AV LYDSVINGNINGER

Lydsvingninger (og andre svingninger) kan matematisk meget elegant beskrives dersom man

anerkjenner at det finns noe som heter  $\sqrt{-1}$ .

For oss dødelige, selv med litt forståelse for matematikk, virker dette umulig.

”Roten” av et tall betyr jo: ”Hvilket tall vi skal gange med seg selv, for å få tallet”

$\sqrt{4}=2$ , fordi  $2 \times 2=4$

$\sqrt{2}= 1,41421356$

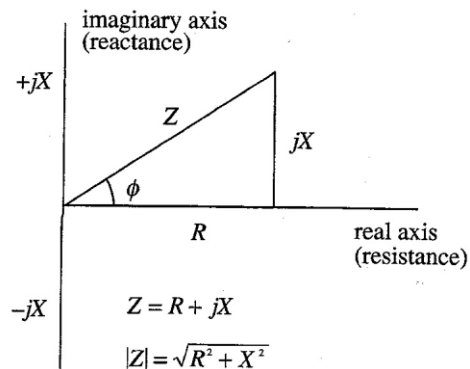
$\sqrt{1}=1$ , fordi  $1 \times 1=1$

Roten av negative tall er egentlig ikke definert!!!

Dere behøver ikke å forstå dette, men vi får en nydelig matematikk ved å ”bekjenne” at  $\sqrt{-1}$  er mulig. Da blir ”utover-aksen” vanligvis angitt som x-aksen, den ”reelle akse”, og ”oppover-aksen/y-aksen” blir den imaginære aksene.  $\sqrt{-1}$  benevnes  $j$  eller  $i$ . Derfor er

$$j^2 = i^2 = -1.$$

Dette gir muligheten for at svingninger kan beskrives som ”vektorer”, dvs de har både en amplitude og en retning.



Svingninger kan beskrives enkelt og beregnes elegant med disse komplekse tall.

Dessverre er det flere alternative notasjonsmåter. Når dere evt. kommer bort i slike notasjoner i bøker etc., husk at  $\omega = 2\pi f$ .

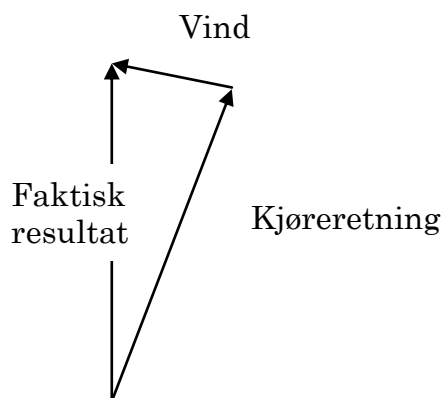
Dessuten blir  $e^x$  alternativt skrevet som  $\exp(x)$  i noen ingeniørkretser (Dette likes ikke av ”ordentlige matematikere”!)

$$e^{j\omega t} = e^{j2\pi f t} = \exp(j\omega t)$$

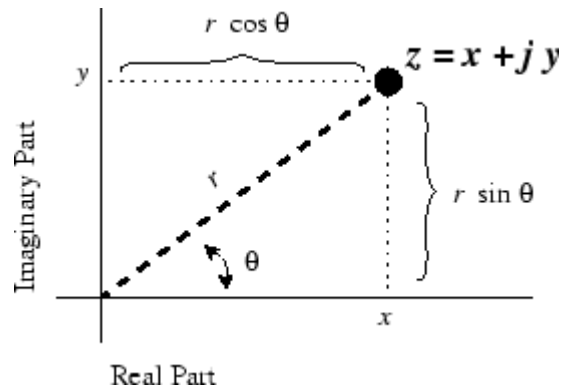
$$= \cos \omega t + j \sin \omega t$$

$$= \cos 2\pi f t + j \sin 2\pi f t$$

En enkel anskueliggjøring av en vektor er bevegelsen av et skip med sidevind.



En VEKTOR kan altså defineres slik:



En summering av 2 sinusoider, med tidsforsinkelse  $\delta$ , kan beskrives som:

$$a_1 \cos(2\pi f t) + a_2 \cos(2\pi f t + \delta t))$$

*jmf Steiglitz s. 15*

eller, hvis vi tenker vektorer:

$$a_1 e^{j2\pi f t} + a_2 e^{j(2\pi f + \delta)t}$$

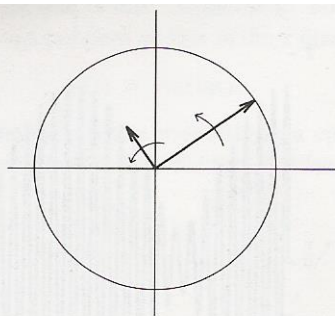
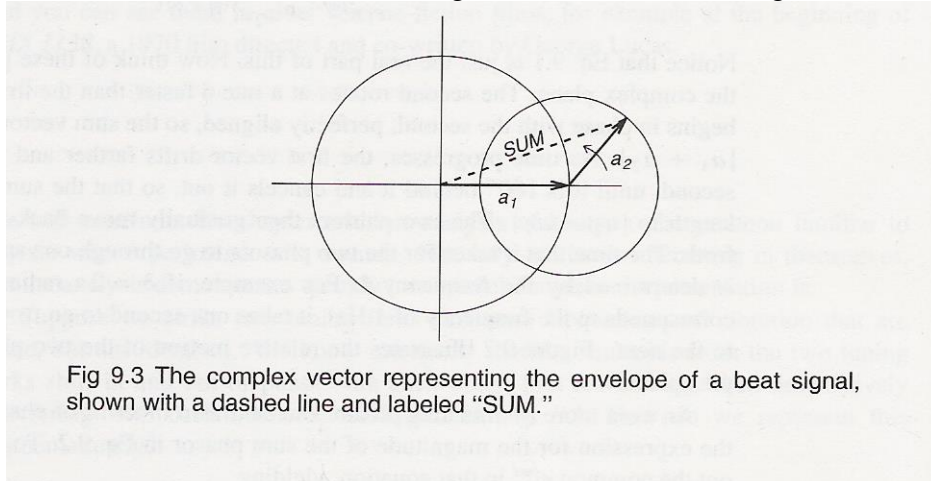


Fig. 9.2 Two phasors with different frequencies. They alternately line up and cancel out.

$$e^{j\omega t} [a_1 + a_2 e^{j\delta t}] \quad (9.3)$$

Hvis vi ser dette som en summering av 2 vektorer, får vi følgende:



Denne figur kan likne på hvordan et "stag/("cam") driver et lokomotiv-hjul. Hvis vi betrakter summen av de to vektorer som en kompleks vektor som roterer med varierende lengde og hastighet, kan vi definere den varierende SUM- lengde som ENVELOPEN av det summerede signal, og dets varierende vinkel-hastighet/"angular speed" som dets frekvens.

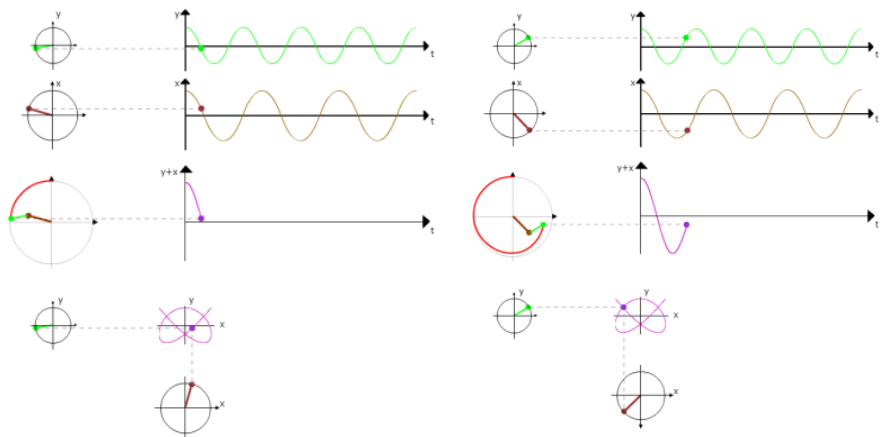
Da envelopen og frekvensen varierer med tiden, må vi benytte begreper som "*instantaneous envelope*" og "*instantaneous frequency*".

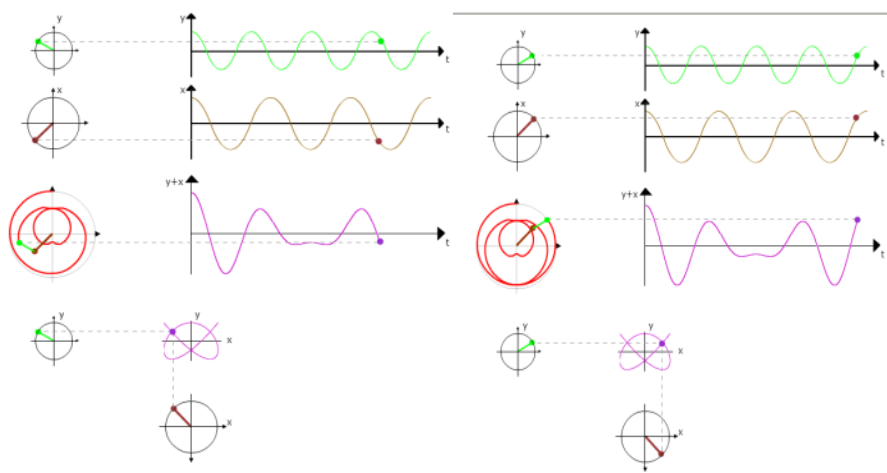
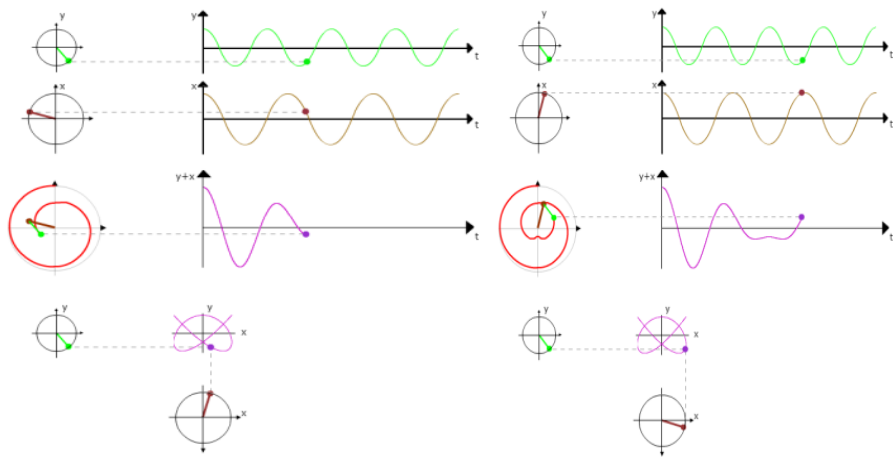
En liten demonstrasjon, tatt fra:

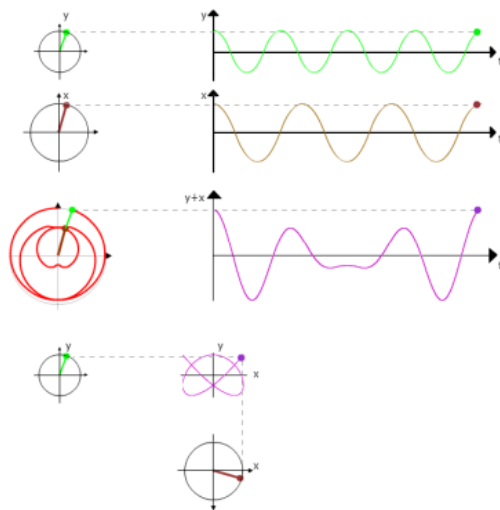
<http://www.animations.physics.unsw.edu.au/jw/phasor-addition.html>

Her ser vi hvordan kurveformen for summen av to sinuser med forskjellig frekvens og fase dannes (i nederste figur i hvert bilde), etter som tiden går (og de to tonene svinger rundt sirkelen i hvert sitt tempo)

F=3/2





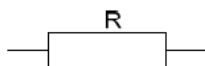


Vektorer benyttes også i (data-)grafikk, da objekter som har størrelse og retning blir lettere skalerbare og endringsbare mhp retning/synsvinkel etc.

# ELEKTRISKE KRETSER, OSCILLATORER, FILTRE ANALOGIER MED MEKANISKE SVINGNINGER

En vanlig elektrisk krets er bygd opp av tre typer komponenter:

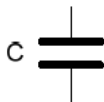
**Motstander:**



Når det går en strøm  $I(t)$  gjennom en motstand med resistans  $R$ , er spenningen over motstanden

$$U_R(t) = R \cdot I(t).$$

**Kondensatorer:**



Når en kondensator med kapasitans  $C$  lades med en strøm  $I(t)$ , er spenningen over kondensatoren

$$U_C(t) = \frac{1}{C} \int_0^t I(\tau) d\tau.$$

**Spoler:**



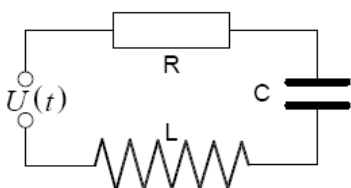
Når en strøm  $I(t)$  går gjennom en spole med induktans  $L$ , er spenningen over spolen

$$U_L(t) = L \cdot \frac{dI(t)}{dt}.$$

**KONDENSATOR Kapasitans** kan vi assosiere med en **spiral fjær**. Vi lader kondensatoren = vi trekker ut fjæra (positiv spenning) eller stuver den sammen (negativ spenning). En høy kapasitet tilsvarer en slapp fjær. Når vi lar fjæra gå tilbake får vi energien tilbake.

**SPOLER Induktans** kan vi se på som en **masse** målt i kg. Vi tilfører den energi ved å flytte den høyere opp i rommet. En høy induktivitet tilsvarer en stor masse. Det er ikke energiforbruk ved dette; når vi senker massen får vi energien tilbake.

**MOTSTAND Resistans** tilsvarer friksjon. Det er bare ved friksjon vi kan tilføre omgivelsene varme.



Vi skal konsentrere oss om en *seriekrets* der en motstand, en kondensator og en spole koples i serie sammen med en spenningskilde som gir en spenning  $U(t)$ . Da sier Kirchhoffs 1. lov at summen av spenningsfallene i kretsen (regnet med fortegn) skal være lik null:

**Resonans** mellom induktans og kapasitans tilsvarer det å henge massen i fjæra (som på den andre siden er fast til en referanse) og så gi massen en dult. Det er ved svingningene som oppstår at kraft og hastighet skifter periodisk på. Frekvensen øker for mindre masser og stivere fjærer, altså mindre induktiviteter og mindre kapasiteter.

Kompleks Impedans:  $= R - j \frac{1}{\omega C} + j \omega L$

## Faseskift til en kondensator og spole.

Spenningen over en kondensator og strømmen ”gjennom” den er ikke i fase. I selve kondensatoren er strømmen faseforskjøvet 90° foran spenningen. Dette kan en forstå når en husker på at kondensatoren må lades opp, dvs. det må gå en strøm, *før* vi får ladning på den, og dermed spenning over den.

Med en sinusformet vekselspenning over en kondensator blir sammenhengen mellom

spenning og strøm som vist i figur 2a. Når strøm og spenning er kjent, kan en også beregne effekten  $P = VI$  som kondensatoren tar opp (se figur 2b), og vi ser da at i løpet av en hel periode  $T = 1/f$  vil kondensatoren levere fra seg akkurat like mye energi som den har tatt opp, dvs. en kondensator forbruker ikke energi.

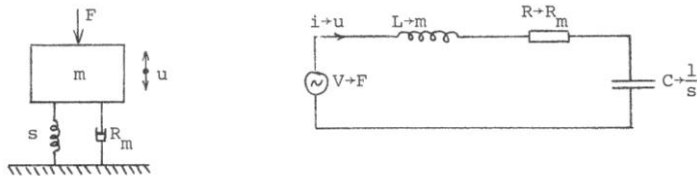
Påtrykkes en *spole* en ytre spenning, svarer spolen med å indusere en spenning som forsøker å nøytralisere den ytre spenningen slik at det bare blir minimale strømdringer per tid. Vi kan altså endre spenningen raskere enn vi kan forandre strømmen. Dersom en sender vekselspenning inn på en spole, vil derfor strøm og spenning være faseforskjøvet i forhold til hverandre, og strømmen vil komme *etter* spenningen.

Resonansfrekvens:  $f_0 = \frac{1}{2 \cdot \pi \cdot \sqrt{L \cdot C}}$  eller:  $\omega_0 = \sqrt{\frac{1}{L \cdot C}}$

## ANALOGI, MEKANISK

Kraft $F$	→	Spenning $V$
Hastighet $u$	→	Strøm $I$
Resistans $R_m$	→	Resistans $R$
Masse $m$	→	Induktivitet $L$
Kompleans $\frac{1}{s}$	→	Kapasitet $C$
(s - stivhet)		

## SVINGEKRETS



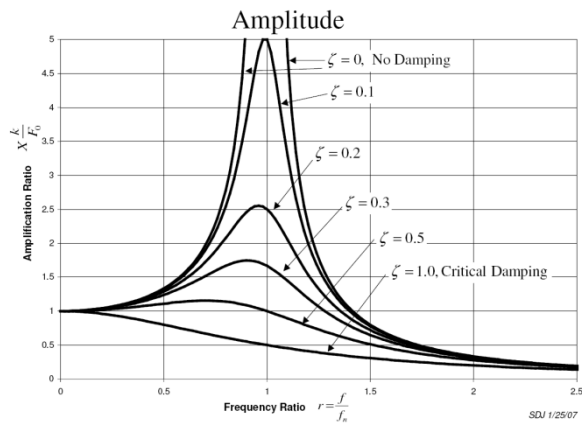
Res Freq:  $f_n = \frac{1}{2\pi} \sqrt{\frac{k}{m}}$ , der  $k$ =fjærstivhet [N/m], og  $m$ =massen [kg]

Enklere, praktisk formel:  $f = \frac{15,8}{\sqrt{d}}$ ,

der  $d$ =den statiske nedbøyning av fjæra ved belastning med massen  $m$



## VIBRASJONSISOLERING (for frekvenser godt over resonansfrekvensen)



Eks. Trommel i vaskemaskin:

-Start: lav frekvens, står stille

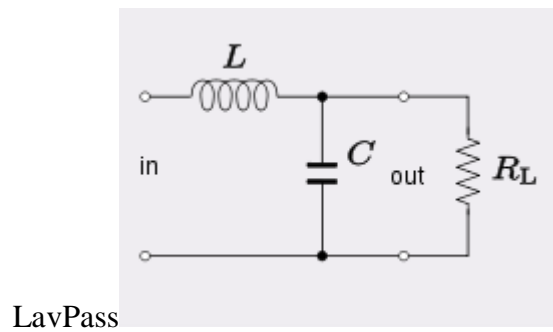
-Litt høyere turtall: Resonans, rister fælt

-Høyere turtall: Står stille, og er Vibrasjonsisolert fra gulvet

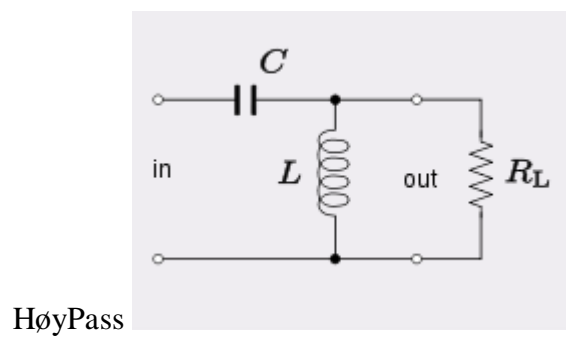
# ELEKTRISKE FILTRE

## FORSKJELLIGE GRUNNTYPER

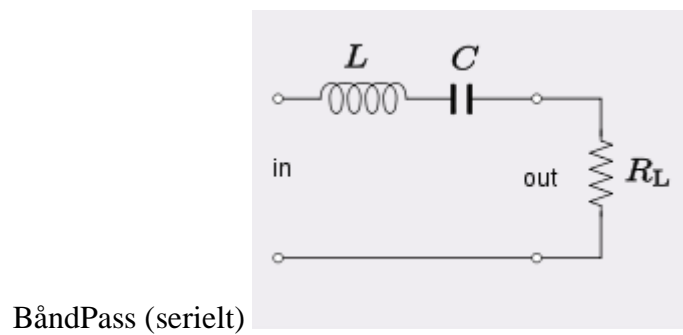
### Lavpass



$$\omega_c = \frac{1}{\sqrt{LC}}, \text{ dempningsfaktor: } \zeta = \frac{1}{2R_L} \sqrt{\frac{L}{C}}$$



knekkfrekvens (-3 dB):  $\omega_c = \frac{1}{\sqrt{LC}}$

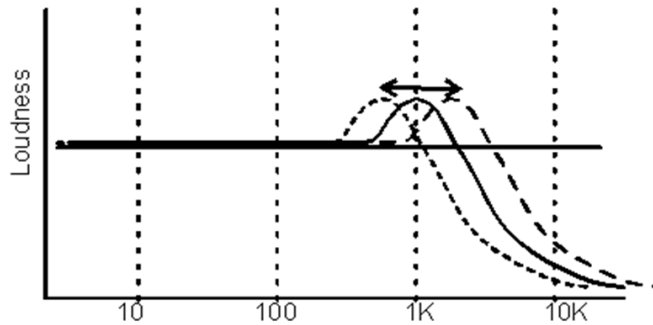


$$\omega_c = \frac{1}{\sqrt{LC}}, \Delta\omega = \frac{R_L}{L}$$

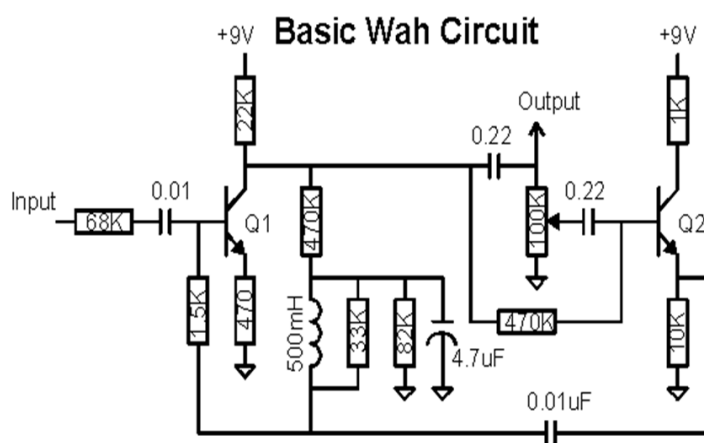
# WAH WAH PEDAL

a bandpass filter or an overcoupled lowpass filter that exhibits a resonant peak just at its lowpass rolloff frequency.

## Wah Frequency Response



Copyright 1999 R.G. Keen



Copyright 1999 R.G. Keen

## ”Nuller og poler” for filtre. ZERO-POLE

Overføringsfunksjonen (”Transfer function”) for et system, f.eks. et filter, kan skrives:

$$X(z) = \frac{P(z)}{Q(z)}$$

der:

- $z_i$  slik at  $P(z_i) = 0$ , er Nullene (”zeros”) for systemet
- $z_j$  slik at  $Q(z_j) = 0$ , er Polene for systemet

I et diagram angis Polene med en  $x$ , mens Nullene angis med en  $o$ .

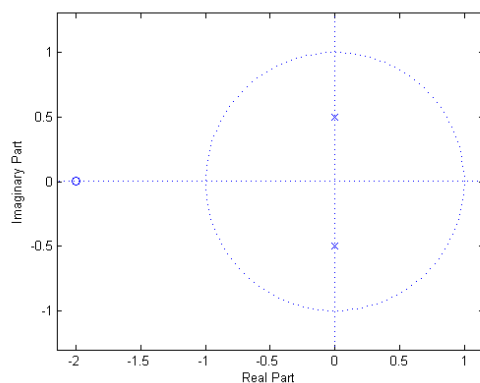
Hvis  $P(z)$  og  $Q(z)$  er faktorisert, kan løsningen plottes I  $z$ -planet.

F.eks., dersom overføringsfunksjonen er:

$$X(z) = \frac{z + 2}{z^2 + \frac{1}{4}}$$

får vi kun én Null, ved  $-2$  og to Poler ved  $\pm \frac{1}{2}$

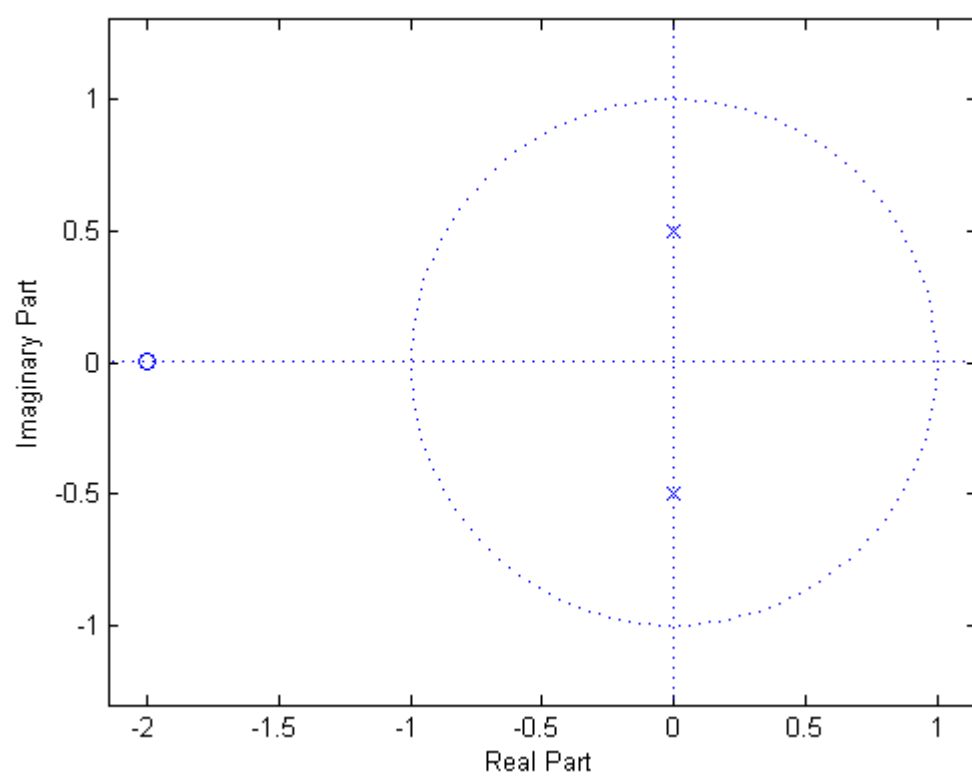
Figuren ser da slik ut:



Et enkelt program som enkelt viser hva slags filtre vi får ved å plassere Zero's og Pole's i det komplekse plan fins på internettsiden:

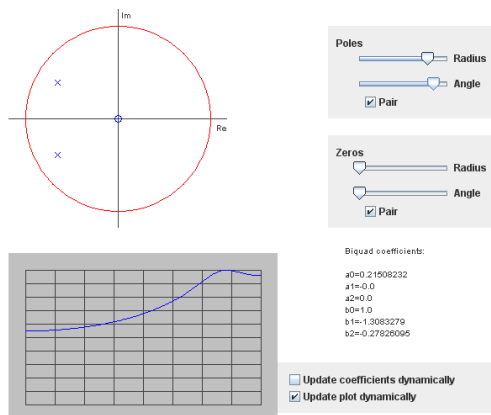
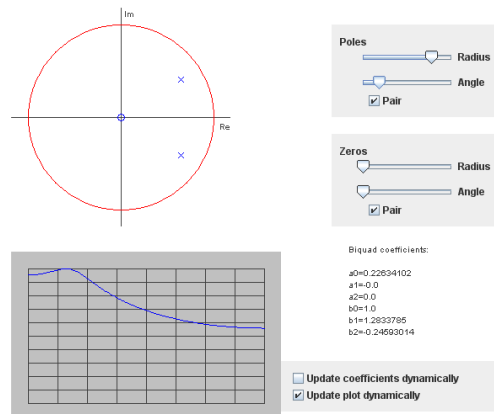
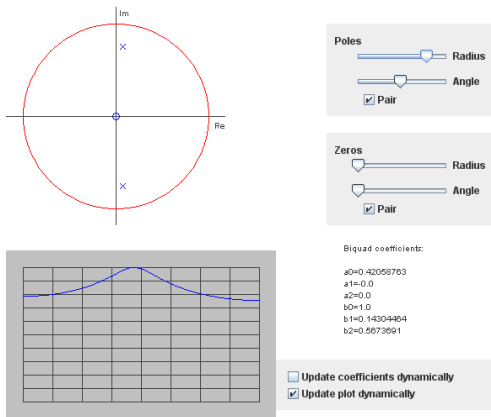
<http://www.earlevel.com/main/2003/02/27/pole-zero-placement/>

Der ser man også filterkoeffisientene vi drøftet tidligere, så dette er to sider av samme sak



# FILTRE, ut fra ZEROs and POLEs

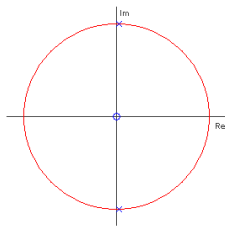
<http://www.earlevel.com/main/2003/02/27/pole-zero-placement/>



- February 2003
- August 2002
- January 1997
- November 1996
- October 1996
- August 1996
- July 1996

## Categories

- Digital Audio (23)
  - Aliasing (6)
  - Convolution (2)
  - Dither (3)
  - Filters (11)
    - FIR Filters (4)
    - IIR Filters (6)
  - Fourier (2)
  - FFT (2)
  - Impulse Response (4)
  - Jitter (2)
  - Phase (2)
  - Reverb (1)
  - Sample Rate Conversion (7)
  - Sampling Theory (1)
- MIDI (2)
- Uncategorized (1)



**Poles**

Radius

Angle

☒ Pair

**Zeros**

Radius

Angle

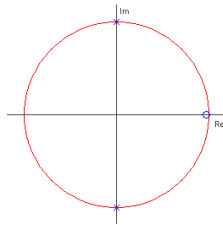
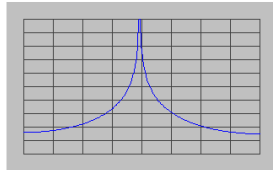
☒ Pair

Biquad coefficients:

a0=0.015465252  
a1=0.0  
a2=0.0  
b0=1.0  
b1=0.082821515  
b2=0.9980267

☐ Update coefficients dynamically

☒ Update plot dynamically



**Poles**

Radius

Angle

☒ Pair

**Zeros**

Radius

Angle

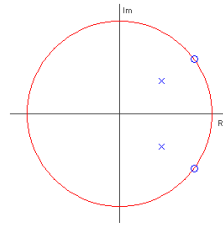
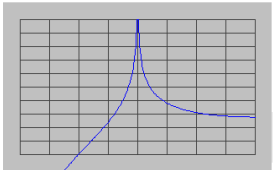
☒ Pair

Biquad coefficients:

a0=0.008070040  
a1=0.016656890  
a2=0.0075931093  
b0=1.0  
b1=1.2246469E-16  
b2=1.0

☐ Update coefficients dynamically

☒ Update plot dynamically



**Poles**

Radius

Angle

☒ Pair

**Zeros**

Radius

Angle

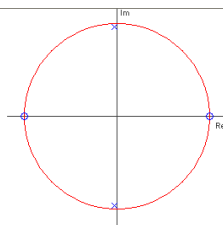
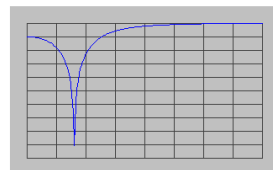
☒ Pair

Biquad coefficients:

a0=0.6227083  
a1=-1.0075632  
a2=0.19242747  
b0=1.0  
b1=0.61667984  
b2=-0.08365928

☐ Update coefficients dynamically

☒ Update plot dynamically



**Poles**

Radius

Angle

☒ Pair

**Zeros**

Radius

Angle

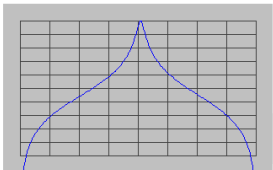
☐ Pair

Biquad coefficients:

a0=0.03889355  
a1=0.0  
a2=0.03889355  
b0=1.0  
b1=0.060308558  
b2=0.91978145

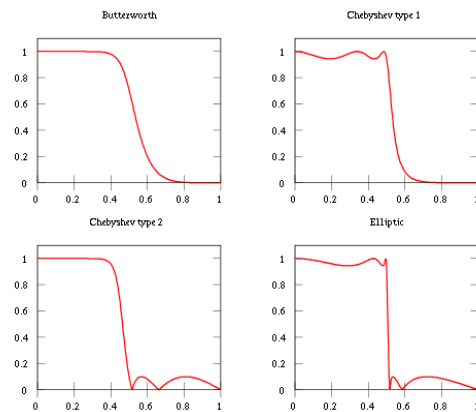
☐ Update coefficients dynamically

☒ Update plot dynamically



Filter-design er en egen ”kunststart”, som vi ikke skal gå inn på i denne bok. Det fins svært mange type filtre som gjør nesten samme jobb. Alle har en eller annen uønsket egenskap, at de ikke skjærer skarpt nok, at de gir noen feil ”krusninger”/(eng: "ripples") i frekvensområder der de ikke skulle gitt signal i det hele tatt, eller at fasen ikke er lik for alle frekvenser (faselineært).

Følgende figur viser et “discrete-time” Butterworth filter, og tre andre vanlige filtertyper: Chebychev type 1 og 2, og Elliptisk. Alle disse filtrene er av femte orden.



Butterworth filteret faller av saktere rundt “cut-off”-frekvensen enn de andre filtrene, men ute deres små “krumninger”/”eng: “ripples”.



## FASE-FEIL i FILTRE

Ofte glemmer man fasen når man beskriver et filter. Det kan medføre en del uheldige konsekvenser. I mange tilfelle kan det hende at fasen ikke er lik for alle frekvenser (f.eks at diskanten faktisk kommer litt "for sent" i forhold til bass/mellomtone). Slike feil fins i mange "gode" filtre. Feilen kan reduseres f.eks ved Oversampling.

## OVERSAMPLING

Ofte glemmer man fasen når man beskriver et filter. Det kan medføre en del uheldige konsekvenser. I mange tilfelle kan det hende at fasen ikke er lik for alle frekvenser (f.eks. at diskanten faktisk kommer litt "for sent" i forhold til bass/mellomtone). Slike feil fins i mange "gode" filtre. Feilen kan reduseres f.eks. ved oversampling

### **Oversampling**

Oversampling med en faktor  $L$ , reduseres Kvantiseringsstøyen med faktor  $L$ .

Å sample med dobbel så høy samplingsfrekvens som nødvendig ( $L=2$ ) har tilnærmet samme virkning som å legge til 1 bit, og dermed økes SQNR (Signal-to-Quantization error Noise Ratio) med omtrent 6 dB.

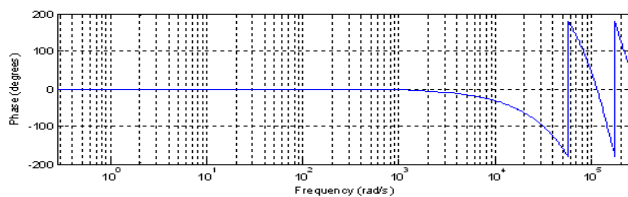
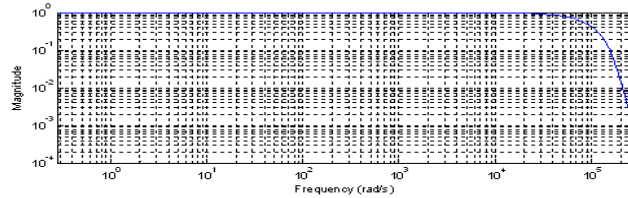
*Hver gang vi dobler samplingsraten, kan vi derfor fjerne 1 bit fra encoderen.*

Dette fordi kvantiseringsstøyen jo spres over hele frekvensområdet opp til halvparten av (den nå økte) samplingsraten, slik at halvparten blir fjernet ved filtrering i det digitale domene. Den andre fordelen ved oversampling er at vi reduserer kompleksiteten av analoge lavpassfiltre, slik at filterets flanker ikke behøver å være så steile, noe som reduserer faseforvrengning.

## Eksempel på oversampling

Merk at de neste kurver er angitt med **frekvensakse [radianer/s]**, dvs  $2\pi f$ , altså  $2 \times 3,14 \times f = \text{ca } 6,3 \times f$ . 20 kHz blir f.eks. 125600 radianer/s (altså noe over  $10^5$ ), og 107 radianer blir 100x så mye, altså ca. 2000 kHz

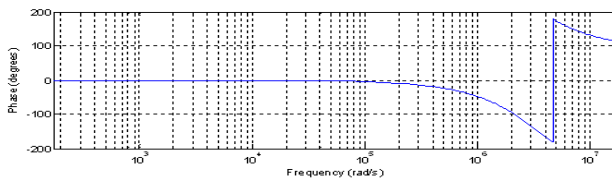
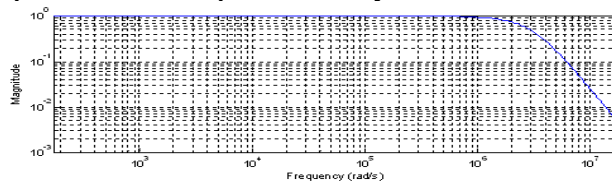
Først benytter vi ”vanlig” sampling, og koster på oss et tiende ordens Bessel lavpassfilter. Vi ser at filteret skjærer veldig skarpt mot høyere frekvenser, så vil skulle tro at dette var bra. Fasen er derimot ikke-lineær mot høyere frekvenser, og det er ugunstig:



Ca 20 kHz

$10^5$  radianer

**Oversampling 64x gir:** Her benytter vi et ”billigere” tredje ordens filter, som vi kan tillate oss, siden det jo kutter av langt over det hørbare området. Det skjærer altså mye mindre skarpt, men fase-responsen er mye bedre. Den er lineær helt fra 0 rad/s opp til 138 544 rad/s.



Ca 2.000 kHz

$10^7$  radianer/s

En ekstrem form for oversampling reduserer kvantiseringen for Analog-Digital Converteren helt ned til 1 bit, er den såkalte **Delta Sigma Konverter**. Det kan virke umulig at vi kan gjenskape en bølgeform som f.eks. en sinus med kun 1 bit, men da må vi huske at vi sampler med en hastighet som er mye høyere enn vanlig.

1 bit tilsvarer jo en firkantpuls, og når vi lavpassfiltrer den kraftig i ved avspilling tar vi bort ”overtoneene”, slik at vi står igjen med en sinus.

### 12.10.1 Måling av filtre i Pure Data (Pd)

Et enkelt filter i Pd er **bp~** som er et **Båndpass-filter**.

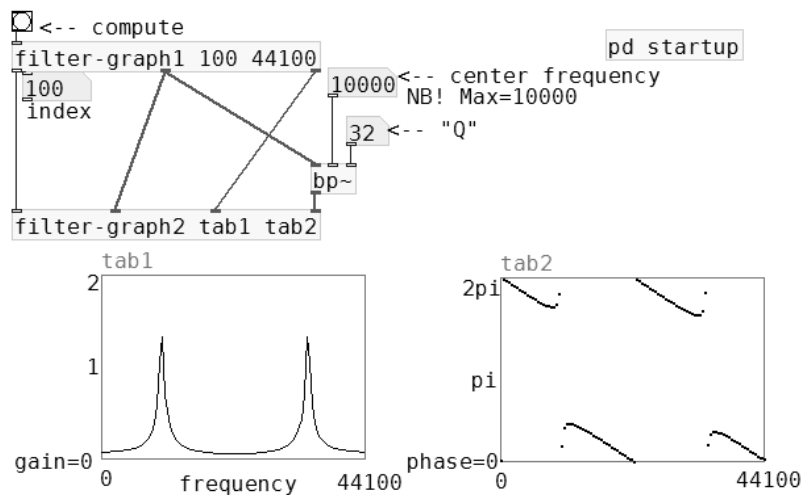
Vi kan stille inn frekvensen, og Q-verdien. For å måle hvordan filteret virker, kan vi benytte abstraksjonene **filter-graph1**, og **filtergraph2**:

#### MEASURING FILTER FREQUENCY AND PHASE RESPONSE

You can use the "filter-graph1" and "filter-graph2" abstractions as shown to test filters. Connect them as shown with a filter between them. Try varying the parameters and/or substituting other filters.

"filter-graph1" takes as arguments the number of points to graph and the frequency range. "filter-graph2" takes as arguments the name of a table to hold the (frequency dependent) gain, and another, if specified, for the phase.

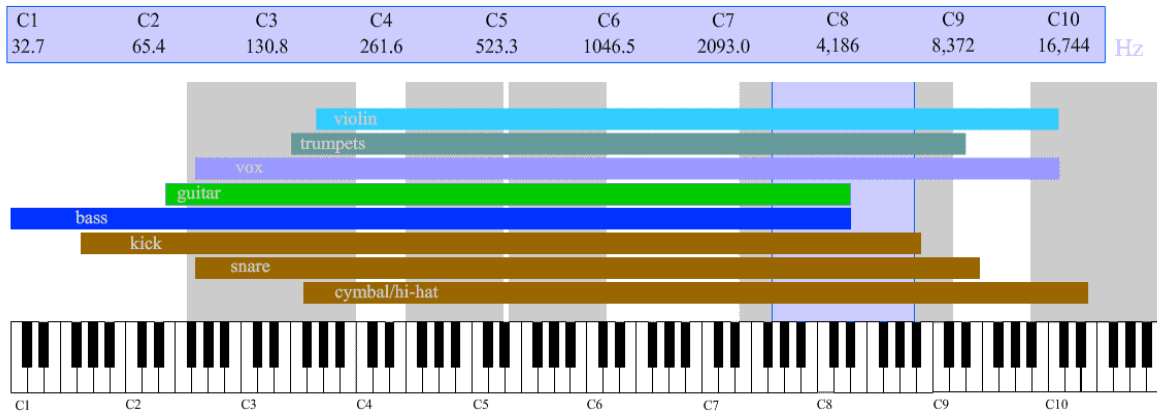
You can edit this patch to replace "bp" with any other filter you're curious about.



Pure Data H10  
updated for Pd version 0.39

## BRUK AV FILTRE

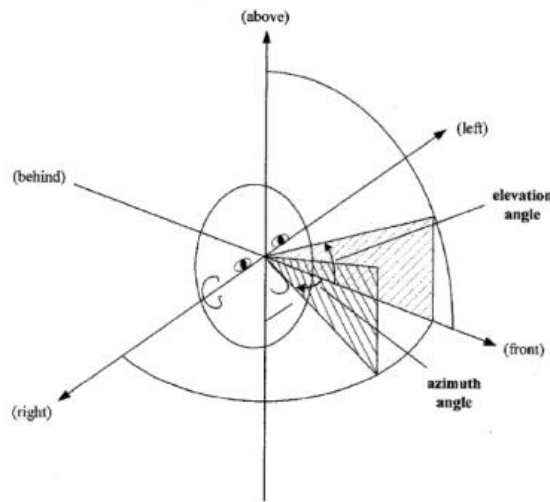
Følgende figur angir hvordan typiske ”problemer” kan justeres med filtrering i forskjellige frekvensområder:



- Romlete (”boomy”) bass ligger ofte rundt 100-250 Hz.
- Man benytter ofte en ”roll off” for frekvensene i området 30-80 Hz.
- Hvis man reduserer nivået v/ ca. 250 Hz, kan det oppfattes som man hadde økt nivået ved ca 5 kHz.
- For mye energi i frekvensområdet 350-600 Hz kan virke ”boksete” (”boxy”/”roomy”).
- I frekvensområdet (600)-1000 Hz kan det være nødvendig å dempe evt. sterke/skarpe områder (”peaks”), også for å gi ”presence”.
- Noe skjærende/skarpe lyder kan opptre rundt 2,5 kHz. Lydbildet blir mykere dersom dette området reduseres, eller livligere dersom man øker nivået for disse frekvenser.
- ”Visle-registeret”/”Sibilance” (3-7 kHz) kan beskrive ”luftighet” i lyd-mixen.
- Støy (”hiss”) lyder opptrer ofte i frekvensområdet 1-4 kHz.
- Frekvensområdet rundt (over) 7 kHz gir ”luft” i lydbildet.

## 6.2.4 HRTF (Head Related Transfer Function)

Definisjoner på retning:

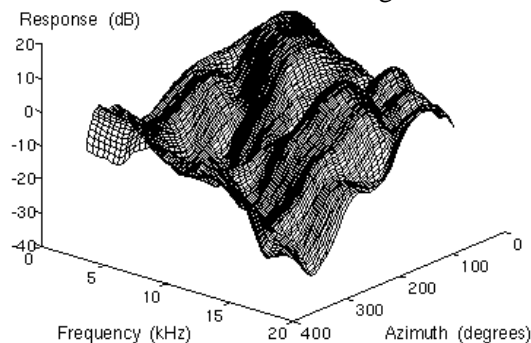


**HRIR (Head Related Impulse Response).** Diffraksjon for overkroppen (“torso”), hodet og det ytre øret gir endringer i lydens utbredelse og oppfattelse, fra kilden til trommehinna. Disse endringer kan sees på som et lineært, tids-uavhengig (time-invariant) filter, som kan karakteriseres med sin Impulsrespons. Det fins da en Impulsrespons for hver retning av lydkilden.

**HRTF (Head Related Transfer Function)** er **HRIR** i frekvens domenet, altså FFT- (Fast Fourier)-analysen av HRIR.

[http://interface.cipic.ucdavis.edu/CIL\\_tutorial/3D\\_HRTF/3D\\_HRTF.htm](http://interface.cipic.ucdavis.edu/CIL_tutorial/3D_HRTF/3D_HRTF.htm)

HRTF må de måles i alle retninger. Her er et eksempel på Horisontal-planet:



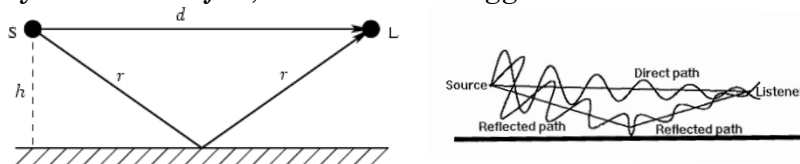
Dette “mesh plot” viser frekvensresponsen for Kemar Dummy Head’s høyre øre, for en lydkilde som beveger seg i horisontalplanet. Selv om overflaten for denne kurve er ganske bulkete, kan vi se en tilnærmet sinusoid endring med endret azimuth. Som forventet er responsen vanligvis størst når kilden er 90° og rettet mot det høyre øret (dvs. rett ut for øret), og svakest når kilden er ved 270° på den motsatte side av hodet.

Vi ser at responsene for front/bak (0° og 180°) er ganske like. Front responsen er noen dB sterkere enn bak-responsen for frekvensområdet 4 til 7 kHz. Peak-verdien ved 4 kHz er tidligere vist til å komme fra resonans i øre-kanalen. “Notch”-verdien ved 10 kHz kommer fra den overnevnte “pinna notch”, og dens frekvens endres med “elevation”.

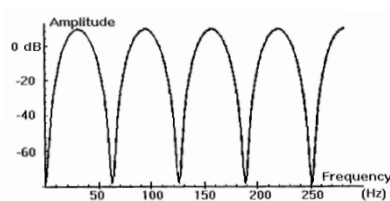
Se også: <http://www.acoustics.hut.fi/research/auralization.html>

# KAM-FILTRE (COMB-FILTERS)

Når et signal kombineres med en tidsforsinket utgave av seg selv, vil noen av frekvensene være i fase i begge signalene, noen vil være i motfase når de møtes, og resten vil være forskjellige steder i mellom. Tidsforsinkelsen kan være rom en fysisk refleksjon, f.eks. fra en vegg:



Prinsippet blir det samme om denne tidsforsinkede kopi blir lagd fysisk ved refleksjoner fra en vegg, i en elektronisk tidsforsinkelses-enhet ("eng.: "delay). Hvis vi ser på en frekvensanalyse av det summerte signal, og benytter en lineær frekvens-skala, vil den typisk se slik ut:



Dette ser jo ut som en kam, derav navnet Kamfilter. Avstand mellom topper (og

$$\Delta f = \frac{1}{\Delta t}$$

mellom bunner) i kamfilteret:

Eks:  $\Delta t = 10\text{ms}$  (gangveiforskjell ca. 3,44m) gir  $\Delta f = 100\text{Hz}$ .

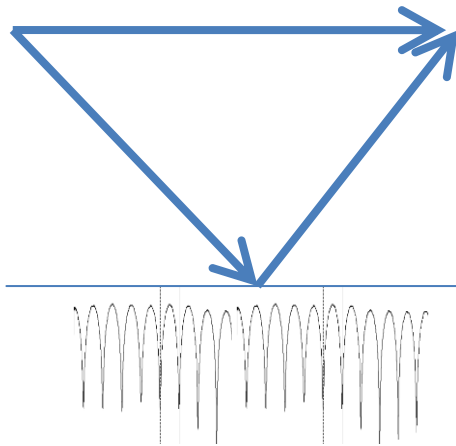
$\Delta f$  kan kalles:

**"Comb-Between-Teeth-Bandwith", eller  
NFI ("Null Frequency Interval").**

Vi ser at formen på filteret ikke likner noen av de vanlige filter-formene. Amn skulle kanskje tro den liknet på en flerbåndet grafisk equaliser, men den har bårndbreddene i logartimisk skala, mens kamfilterene jo har lieært forhold mellom de suksessive "dip'er". Derfor er det ikke enkelt, faktisk nesten umulig, å benytte vanlige filtre for å fjerne kamfilter-effekter. (Man må evt. kompensere med en tidsforsinket signal i motfase, noe som er praktisk noe problematisk).

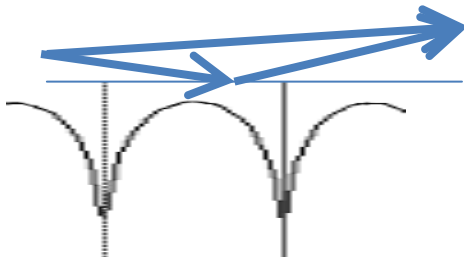
**a) Lang tidsforsinkelse:**

Mange, meget trange kamfiltre. Høres neppe som klangfarge-endring.  
Her er tidsforsinkelsen så lang at det trolig oppfattes som et distinkt ekko i tidsdomenet.



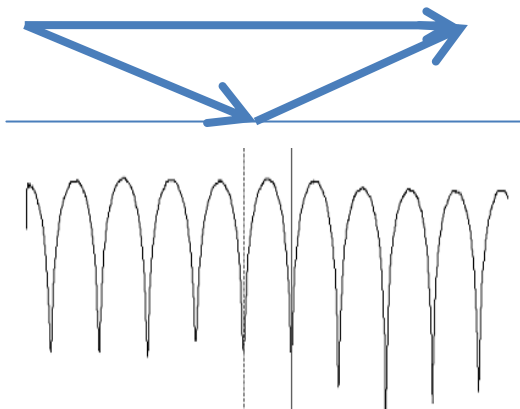
**b) Kort tidsforsinkelse:**

Meget brede kamfiltre.  
(nesten som enkel tone-kontroll-knapp)



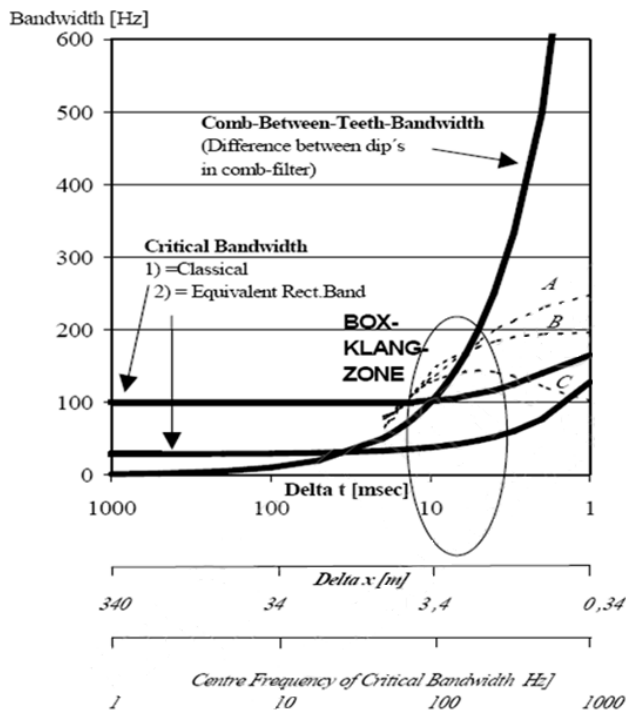
**c) Middels tidsforsinkelse:**

(Eks  $\Delta t=10\text{ms}$ ,  $\Delta s= 3,44\text{m}$ ,  $\Delta f=100\text{Hz}$ )



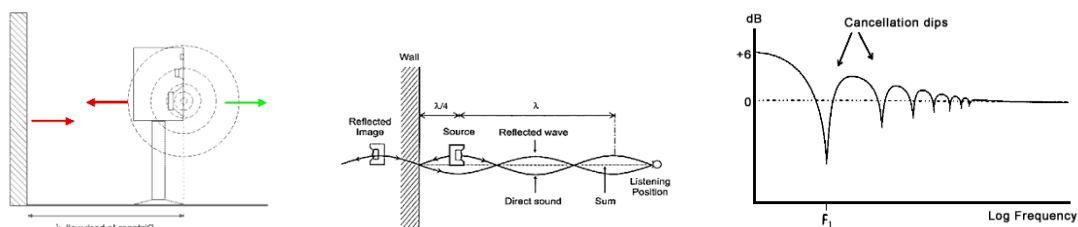
## Kamfiltre og klangfarge (timbre)

For slik moderate tidsforsinkelse h rer vi en klar endring av klangfargen ("timbre"). Dette oppleves sterkest for distinkte refleksjoner som gir et kamfilter der b ndbredden ("Comb-Between-Teeth-Bandwidth") er i st rrelsesorden Kritisk B ndbredde (frekvens-oppl sningen i basil rmembranet i indre  ret). Dette er blitt kalt "Box-Klangfarbe", fordi det l ter som om man st r inne i en slags boks (og det gjør man jo ved slike avstander til vegger).



Box-Klangfarbe er blitt observert p  orkester-podier med plane, reflekterende plater relativt lavt hengende over et orkester, og oppleves daglig i korridorer med plane, reflekterende sidevegger.

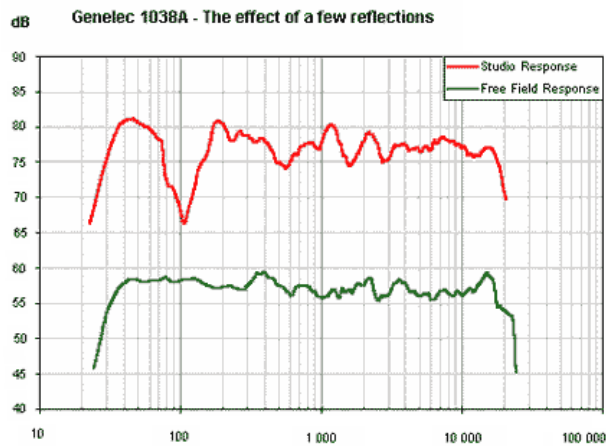
Genelec's hjemmeside angir f lgende mhp frittst ende h yttalere foran en reflekterende vegg:



Dette er akkurat samme "kam-filter"-figur som over, men n  med logaritmisk frekvens-skala i stedet for line r. For   sikre at denne f rste "dip" kommer lavere enn f.eks. 30Hz, m  h yttaleren faktisk plasseres mer enn 2,8m fra vegg.

Et typisk eksempel p  hvordan rommet p virker responsen fra h yttaleren:

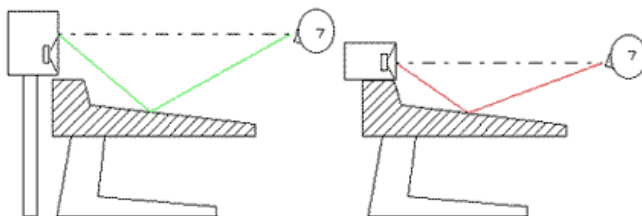




Genelec har til og med utgitt et målebånd, som angir ”frekvens for første kamfilter-dip” i stedet for meter.

Men det er ikke bare vegger som kan gi uheldige refleksjoner i et lydkontrollrom. Refleksjoner fra mixe-bord etc. og høye ”racks” gir ofte kam-filtre. Endret retning/plassering av høyttalere kan kanskje gi noe bedring men er ikke alltid nok til å eliminere denne effekt.

Selv om man sikrer at høyttalere plasseres langt fra reflekterende vegger, er det fremdeles mange flater som kan gi betydelige refleksjoner. (Og jo mer rommet er dempet, jo mer vil slike enkelt-refleksjoner gi merkbare Kam-filtre): Et vanlig problem er refleksjon fra miksebord eller fra andre racks:



## Kamfiltre pga mikrofonplassering.

Man får tilsvarende kamfiltre også ved andre tilfelle der man mikser et signal med en kort tidsforsinkelse av seg selv. Dersom man f.eks. tror man får bedre og sikrere dekning av en person på en talerstol ved å montere 2 mikrofoner, f.eks. én fra hver side, vil man få god lyd kvalitet kun dersom personen står helt nøyaktig stille absolutt midt i mellom mikrofonene, og han så liten munn at alle lyder kommer ut av ett punkt. Ellers vil man få en liten tidsforsinkelse mellom mikrofonene, slik at man får kamfiltre og en ”boksete” lyd kvalitet. Dersom taleren flytter seg sideveis, vil kammene/”nullene” i filteret flytte seg oppover/nedover, nesten som en wah-wah-pedal eller flanger.

Dersom man har en anordning i mixeren som kun velger den sterkeste mikrofonen, blir det selvfølgelig ingen kamfilter-effekt. Det beste er dog trolig å benytte en såkalt automatisk nivåkontroll (”Automatic Gain Control”, ”ACG”) for slike vimsete og ujevne talere.

Kamfiltre får man også ved å plassere en mikrofon i en (kort) avstand fra en reflekterende flate, f.eks. 3-4 cm over en bordflate, eller tilsvarende under en reflekterende himling. Dersom man plasserer mikrofonen med en meget liten avstand fra flaten, nesten i flukt, blir dog kamfilteret så bredt at det ikke gir noen uheldig innvirkning. Dette benyttes i såkalte PZM Pressure Zone Mic

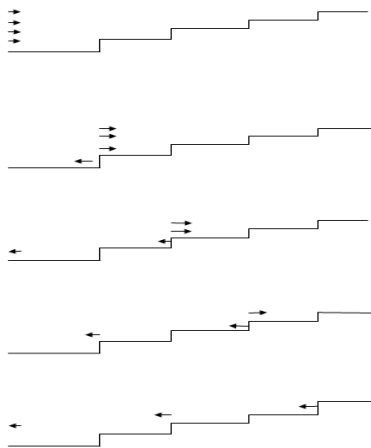
Noen ganger er slike **kamfiltre** ikke til å unngå, f.eks. fra gulvet når en person taler/synger med mikrofon på stativ. Man kan selvfølgelig legge et teppe under mikrofonstativet, men det er jo ikke alltid ønskelig, rent praktisk. Bruk av retningsmikrofoner og ”nærmikrofoner” er selvfølgelig en mulighet for å redusere kamfiltereffektene. Noen ganger er slike kamfiltre ønskelig, f.eks. dersom man ved trommeopptak ønsker å få litt **”ambience”** i trommelyden.

Kamfiltre er også grunnlaget for effekter som **FLANGER** etc. og den fargingen/”nærheten” **Elvis Presley** hadde på flere av sine tidligere innspillinger kom fra at sangeren ble plassert i en korridor, med korte refleksjoner fra sideveggene i RCA-studioene i Memphis, visstnok som et forsøk på å etterlikne den effekten Sam Phillips brukte med to spolebåndspillere som ga en liten tidsforsinkelse, i Elvis’ sine aller første innspillinger i Sun Studios. PS! Den noe lengre tidsforsinkelsen man får i lengderetningen i en korridor, stemmer mer med den noe lengre tidsforsinkelsen, som nærmer seg et distinkt ekko, såkalt Elvis-ekko.

Kamfiltre ble også benyttet til å lage **”liksom-stereo”** fra monoinnspillinger.

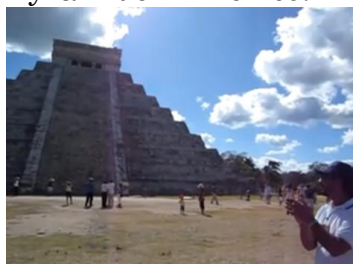
## Kamfiltre fra repetitive refleksjoner. Trapper, pyramider, korridorer

Dersom vi har en bredbåndet lydkilde (f.eks. lyden av en vann-fontene), og får én refleksjon, vil vi kunne høre en klangfarge-endring. Vi kan til og med "spille" f.eks. durskalaen ved å endre avstanden fra den reflekterende "veggen". (Hvis det hadde vært praktisk mulig å gjøre så raskt at vi ikke hadde glemt den forrige "tonen". Dersom vann-fontenen står foran en (ikke alt for høy) trapp med jevne trinn, vil det komme rytmiske refleksjoner med samme tidsforsinkelse. Dette vil gi en forsterket kamfilter-effekt og en tone som er avhengig av trinndybden. Dette kalles **"Repetition Pitch"**.



Slike rytmiske refleksjoner kan oppstå også f.eks. mellom parallelle, reflekterende vegger (flutter ekko/"ping-pong-ekko"), eller fra høye stolrader med reflekterende plate øverst (f.eks. i Oslo Konserthus, men kun uten publikum i stolene), og for så vidt også fra rader i amfier, også greske amfiteatre. De er dog hørbare mest tilbake til lydkilden (klapperen/taleren/scenen), og er ikke grunnen til den anerkjente akustikk til tilhørerne i greske amfiteatre.

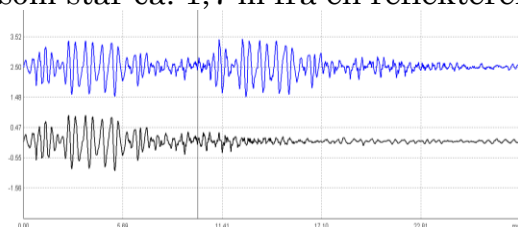
Dersom trappa var meget høy, blir det hele litt mer komplisert, og enda morsommere. Hvis vi står på bakken, vil den effektive trinndybden da øke merkbart oppover i trappa. Da vil tonehøyden for kamfilteret reduseres etter hvert som refleksjonene kommer fra trinnene høyere og høyere opp i trappa. Dette har skapt overtro ved pyramider i Sør-Amerika, der en klapping gir en nedadgående glissando, som høres ut som en guddommelig fugl, eller en klapperslange, avhengig av dimensjonene på to pyramider ved Chichen Itza Pyramiden i Mexico.



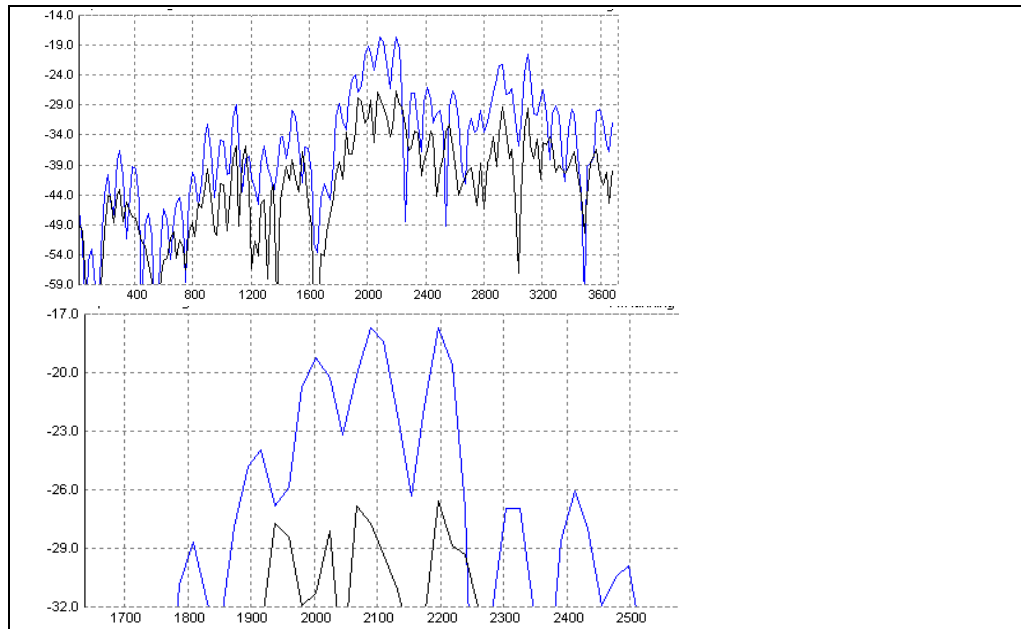
## Blindes ekkolokalisering

Blinde kan faktisk lokalisere relativt godt vha. hørselen. (De har lang trening). For middels/lange avstander er dette forholdsvis enkelt å forstå, da man jo får distinkte ekko (for refleksjoner som kommer f.eks. mer enn 50 ms etter direktelyden, for de aller fleste ”vanlige” lyder.) For reflekterende flater/objekter nærmere, er det noe mer imponerende at blinde kan lokalisere såpass godt vha. ørene. Bredbåndede signaler som vindstøy, støvler mot grus, skraping med blinde-staven etc gir kamfilter-refleksjoner og endring av klangfarge (”timbre”), og fargingen er, som vist over, avhengig av avstanden. Blinde benytter også korte klikk med munnen som kilde for ekkolokalisering. Noen av disse er meget korte, og egentlig ikke så bredfrekvente, og derfor benytter de blinde ofte trolig både ”frekvensplanet” (med klangfarge-endringer av bredfrekvente støysignaler) og tidsplanet (selv 2ms tidsforsinkelse er faktisk mulig å høre for et meget kort klikk).

Klikkene som benyttes i blindes ekkolokalisering er meget personlige, og varierer fra individ til individ. Et eksempel på impulsresponsen for et slikt klikk, med en refleksjon etter 10 ms (som tilsvarer en gangvei på 3,44 m, altså tilsvarende en person som står ca. 1,7 m fra en reflekterende vegg):

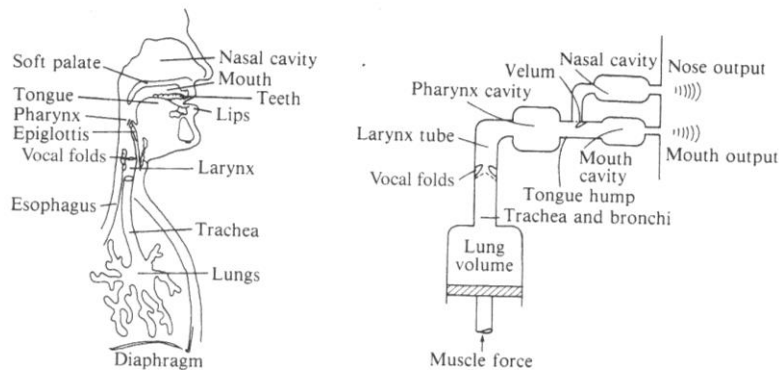


Frekvens analyse med denne refleksjonen (blå) og uten (sort) viser at selv dette korte, ganske høyfrekvente (og dermed ikke så bredfrekvente) signalet, gir en kamfilter (med 100 Hz som ”Comb-Between-Teeth-Bandwidth”, som stemmer med 1/10ms). (Figuren nederst er zoomet inn på det området der signalet, og effekten, er sterkest)



### 2.7.1 Tale. Formanter. Sonogram

Luft fra lungene passerer stemmebåndene. Når disse er åpne, kan vi puste uten å lage særlig mye lyd. Når vi setter dem i vibrasjoner gir de en "buzz"-liknende lyd. Stemmebåndene hos menn vibrerer langsommere enn hos kvinner og barn. (typiske verdier er hhv 110, 220 og 300Hz). Dette gir opphav til de såkalte "stemte" lyder. For ustemte lyder settes stemmebåndene ikke i vibrasjon. (Man kan føle forskjellen mellom stemte og ustemte lyder ved å holde en finger på strupehodet, og veksle mellom lydene "zzzzz" og "sssss".



Den stemte "buzz"-lyden er omtrent lik en trekantbølge, og har dermed mange overtoner. En ustemt "ssss"-liknende lyd er også relativt bredfrekvent. Begge disse lyder kan derfor endres/filtreres ved å endre relativ avstand og form mellom øvre del av ganen ("palate"), tunge, lepper og tenner. Dette gjøres på en meget sofistikert og kompleks måte, som vi lærer ved å herme, og som er vanskelig å beskrive fullt ut. **Konsonanter** kan være stemte eller ustemte, mens **Vokaler** alltid er stemte.

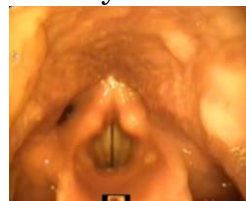
TONE:



PUST:



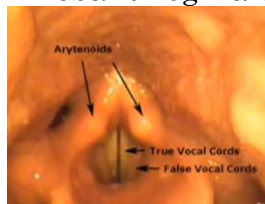
Lys tone:



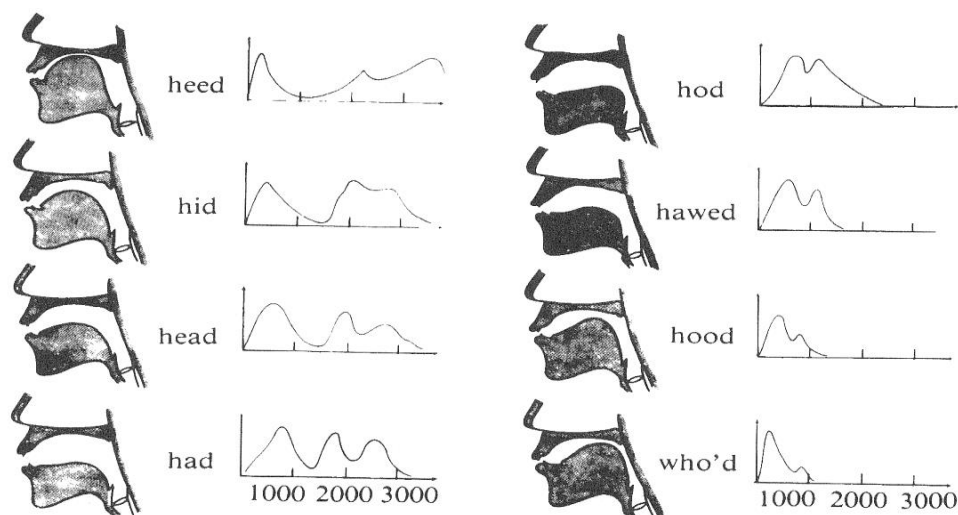
Dypere tone:



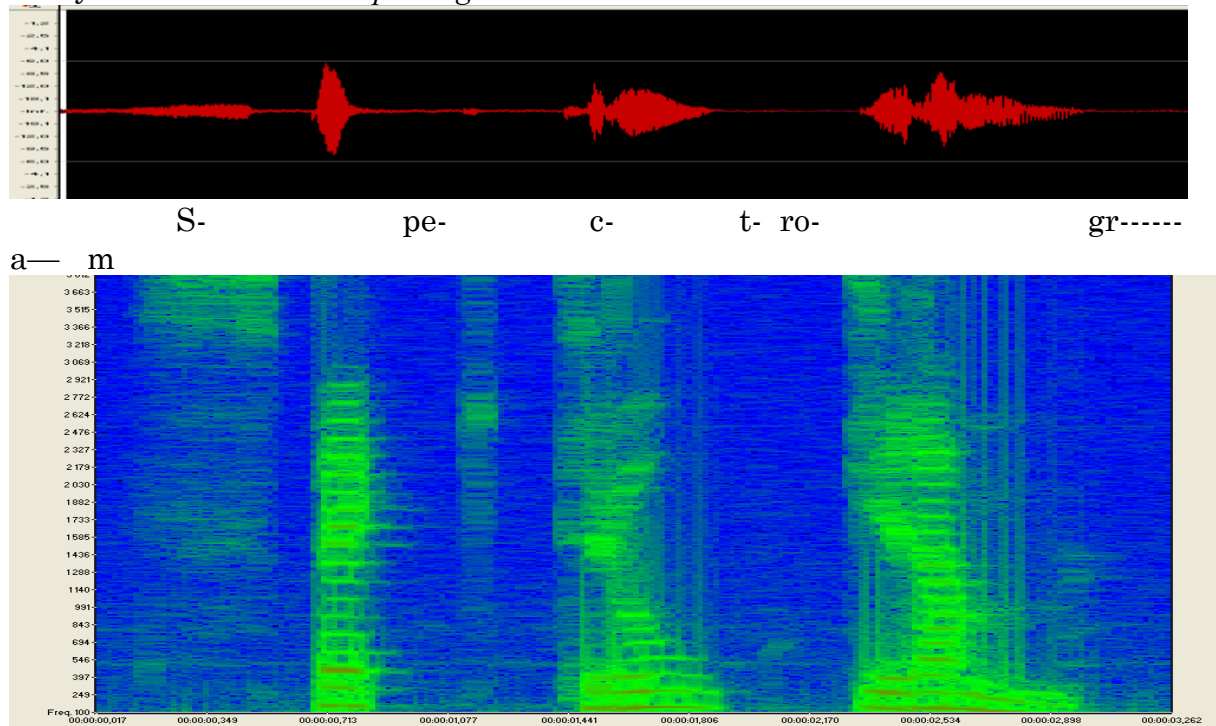
Såkalte “ekte stemmebånd” og “falske stemmebånd”:



Vokal-lyder har ofte frekvenstopper som er uavhengig av hvilken tonehøyde vi snakker (synger). Dette kalles **Formanter**. Disse er altså resonanstopper vi danner ved måten vi plasserer tunge, lepper etc. Følgende figur viser ”røntgenbilde” av munnen når man (med samme grunntone) uttaler forskjellige engelske vokaler fra “heed” til “hood”. (x-aksene er frekvens og y-aksene er dB). Vi ser at frekvensområdene for formantene synker gradvis:



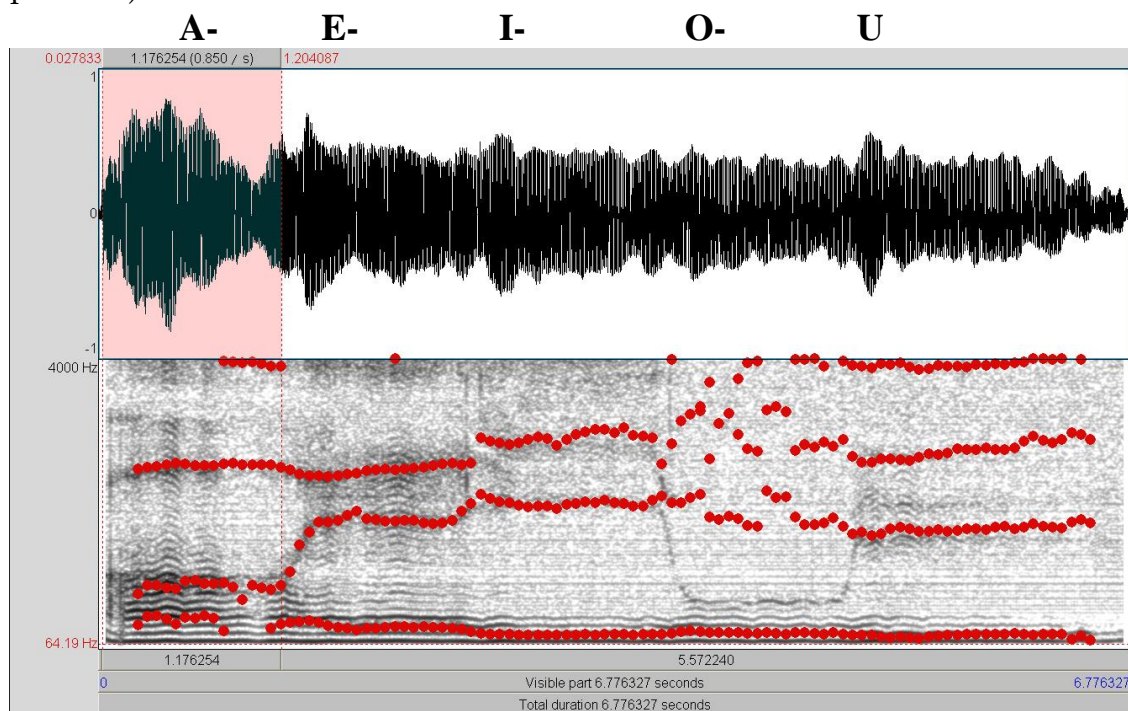
En trøndersk variant er det gamle Arbeiderparti-slagordet: "Æ-E-I A-Æ-Å" ("Jeg er i Arbeiderpartiet jeg også"), som er en rendyrket øvelse i foranter. Analyse av tale gjøres ofte ved hjelp av **spektrogrammer** (eller **sonogrammer**, siden vi tar spekteret bare i det hørbare området). Et vanlig eksempel er en analyse av selve ordet "*spectrogram*":



Mer inngående forklaring av FFT/Spectrogram/Sonogram, se **Feil! Fant ikke referanseilden.** og [Presisjon i Tid og Frekvens](#).

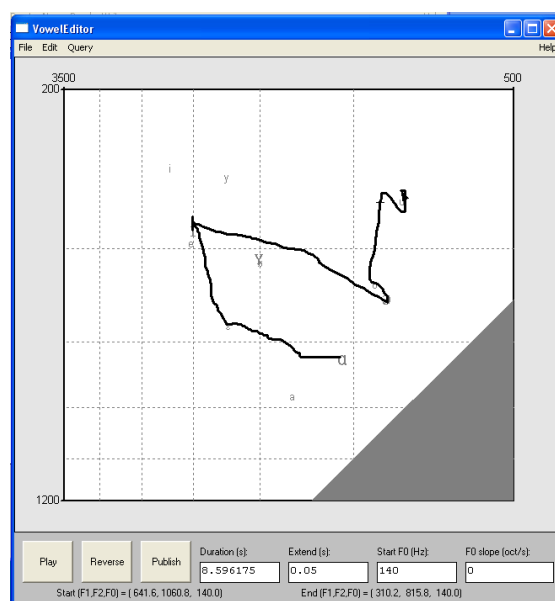


Programmet PRAAT er lagd for analyse (og syntese) av tale, og har gode metoder for å analysere bl.a. formanter. Her er en enkel analyse av herrestemme: A-E-I-O-U: (Øverst selve lydfila, under spectrogrammet og de første formantene (i røde punkter):



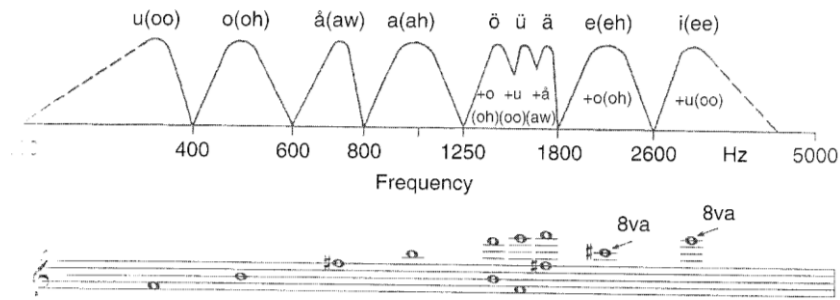
(For analyse av tonehøyde for denne lydsnutt, se [Autokorrelasjon](#))

I PRAAT kan man syntetisere en liknende formant-"melodi" ved å lage en "rundtur" de forskjellige formantene (her igjen: A-E-I-O-U) i den såkalte "*Vowel Editor*":





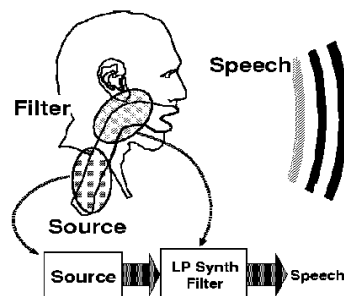
Meyer angir følgende figur over hvordan vokalene kan angis som noter (for Tysk):



2.5 Frequency location of formants for the vowels of the German language (after Thienhaus, Winckel, 1960; Trendelenburg, 1961)

### 15.3 LPC Linear Predicting Code

LPC kommer fra talekommunikasjon. Samplet tale inneholder en mengde data, og blir ofte for “tung” datamessig å overføre på telelinjer. LPC er i utgangspunktet en analyse/resyntese-metode. Man analyserer (tale-)signalet og splitter det opp i **1) Kilde** (Buzzing i stemmebånd eller støy i frikative lyder) og **2) Filteret** som dannes av lufttrøret og munnhulen. Dette er omtrent det motsatte av konvulsjon, altså en de-konvulering. Kontinuerlig tale kan syntetiseres ved å kontinuerlig oppdatere disse parametre omtrent hvert 40 millisekund. Lydkvaliteten av en slik talesyntese er meget mekanisk og lite “menneskelig”, som et leketøy, men den krever liten dataoverføring, typisk kun et par kbits/sekund. Ca 30-50 “frames” pr. sekund gir forståelig tale.



Med LPC kan man manipulere rytme, tonehøyde og klangfarge hver for seg. LPC er videreført i musikalsk sammenheng med at man foretar såkalt Kryss-syntese (“cross synthesis”) der man lar f.eks. klangfargen av en lyd formes av tonehøyde og rytme fra en annen. (Eksempel: “Snakkende stekepanne”)

Det underlige navnet *Linear Prediction* kommer av at man i analysen av spekteret “predikterer” dvs. at vi antar at det kommende sample “følger samme trend” som de forrige (såkalt “ekstrapolering”). Den enkleste antagelsen er at den kommende verdien fortsetter linjen mellom det forrige sample og samplet før det. Mer sofistikerte algoritmer ser også på flere foregående samples. Det antatte/ “predikterte” sample kan selvfølgelig vise seg å være feil, derfor inneholder algoritmen alltid et estimat av feilen. Siden prediktoren betrakter summer og differenser av tidsforsinkede samples, kan vi se på den som et filter (se [Digitale filtre](#)). Hvis vi tar øyeblikks-verdier av disse [filterkoeffisientene](#) over tid, inverterer dem, og siden lar et rikt, bredbåndet signal gå igjennom dette filteret, får vi en rimelig god tilnærming til det tids-varierende spekter i originalsignalet. En “side-effekt” av denne prediksjonen at vi estimerer spekteret av kilden. Dette er en viktig del, men det er bare ett steg i LPC analysen. De andre stegene blir benyttet for tonehøyde, amplitude, og bestemmelsen av om signalet er stemt eller ustemt.

## LPC analyse

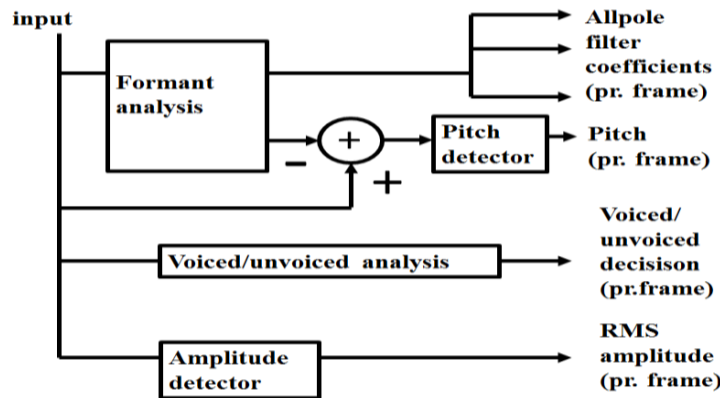
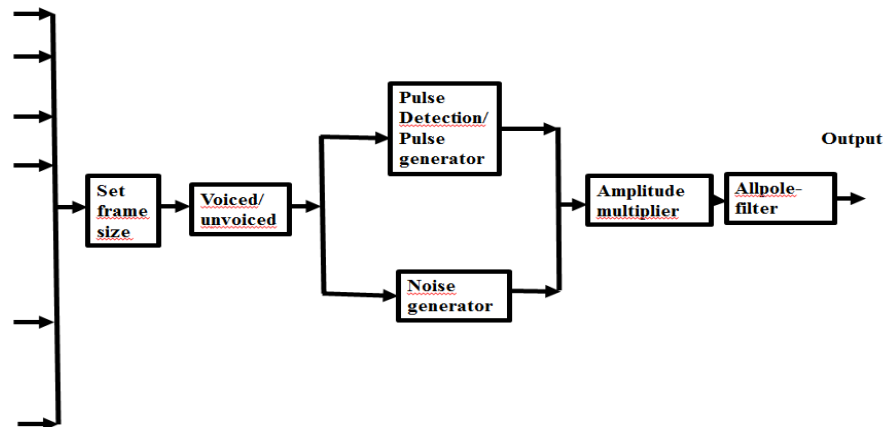


fig: TH



## LPC resyntese

### 1) Formant analysis/Filter estimation

Filtre kan beskrives i et pole-zero-diagram (se [Zero-Pole](#)). Enkelt kan vi si at pole'ne er resonanstopper, dvs. formantene i spekteret, mens zero'ene er nullpunkter ("notches"). Når et filter har flere jevne peak'er kalles det et "allpole" filter. Slike filtre kan modellere spektra som har noen få formant-"peaks", og er en fornuftig tilnærming for mange lyder som blir skapt av menneskestemmen, og visse musikkinstrumenter.

### 2) Pitch og Amplitude-deteksjon

Deteksjon av Tonehøyde ("pitch") gjøres vha. en av metodene beskrevet under [Tonehøyde-deteksjon](#). I summasjons-knutepunktet i figuren over fjerner man formantenes innvirkning fra talesignalet, og estimerer intensiteten og frekvensen for den gjenværende "buzz". Denne fjernings-prosessen kalles "invers filtrering", og det gjenværende signal kalles "residue". Deteksjon av amplitude er en forholdsvis enkel prosedyre. Det inverse av et "allpole" filter er et "all-zero" filter. Det filteret vi finne i analysen skal inverteres for å benyttes i resyntesen i "mottakerenden". Med andre ord: Det inverse filteret kansellerer ut effekten av den spektrale envelopen av signalet. Tilnærmingen er aldri perfekt, slik at det er alltid et "rest"-signal som kalles "residual", som er eksitasjons-funksjonen (en serie av impulser) pluss støy.

### Momenter om LPC

LPC kan sees på forskjellige måter. Enten kan man se på LPC som en prediksjon i tidsdomenet, eller som en de-konvolusjon, der man separerer kilde og filter. Når man har lagd "all-zero"-filteret (noe som er lineær algebra ved løsning av en matrise), kan man enkelt invertere dette til et "all-pole" filter som representerer resonansene i den originale lyden. Siden tale er det antatte "target", antas kilden å være et puls-tog (for "puffs" med luft i stemte talelyder med tonehøyde) eller hvit støy (for luft-turbulens for ustemte konsonanter).

Fordelen med denne analysen er kunnskapen om plassering av formantene. Poler i “all-pole”-filteret gjenspeiler spektrale resonanser. Man må bestemme hvor mange poler man skal benytte når man skal tilpasse filteret til spekteret: For mange eller for detaljerte forkludrer formantene, mens for få gjør at vi ikke kan modellere vokal klangfarge realistisk.

**Kryss-syntese** (“cross synthesis”) er også mulig, ved å kjøre andre eksitasjoner enn tale gjennom filteret. Slik kan man “få en stekepanne til å tale” Dessuten har man gjort tonehøyde uavhengig av klangfarge, slik at tale kan bli sang (ved å legge på en ønsket tonehøyde/“pitch”-envelope), eller sang kan endres i toneomfang (“range”) f.eks. ved å manipulere “raten” for puls-toget for stemme-lyder.

“Windowing” gjør det også uavhengig tids-skala endringer mulige. Dessuten gir LPC-analysen oss også innsikt i selve stemmesignal, f.eks. kan vi skille mellom stemt (“voiced”) eller ustemte “frames” slik at vi kan muliggjøre spesielle transformasjoner, som f.eks. å beholde bare de ustemte konsonantene og ikke re-syntetisere noen tonehøyde overhode.

Algoritmene i LPC likner på [Rekursive Filtre](#), ved bruk av [z-transformen](#) og andre matematiske teknikker.

LPC er ikke noe universalmiddel, fordi lyd kvaliteten ofte er ganske “buzzy”, og den lineære analysen gjør “skarpe forutsetninger” om hvordan lyden som skal behandles, er.

### **Komponister som har benyttet LPC-koding**

F.eks. Paul Lansky: “*Six Fantasies on a Poem by Thomas Campion*” (1978-9) og Charles Dodge: “*Any Resemblance is purely coincidental*” (1980)

Fin side med en del matte: Julius Smith III Stanford CCRMA  
<http://ccrma.stanford.edu/~jos/mdft/>

Med klangfull hilsen!



Tor Halmrast