

## INF 1040

# Sampling, kvantisering og lagring av lyd

### □ Temaer i dag :

1. Analog eller digital, kontinuerlig eller diskret
2. Sampling, kvantisering, digitalisering
3. Nyquist-Shannon teoremet
4. Oversampling, undersampling, aliasing
5. Avspilling av digitalisert lyd
6. Maskering og redundans

- **Pensumlitteratur:** Læreboka, kapittel 11  
NB! Bruk også appendiks A og B !

## Analog

- ”**Analog:** Any form of transmission of information where the transmitted signal’s amplitude or frequency is varied in direct proportion to the intensity of the sound,....”  
*Chambers Dictionary of Science and Technology*

- I analog lagring av lyd er det en direkte proporsjonalitet mellom den virkelige lydintensiteten og det vi lagrer, og den lagrede informasjonen er kontinuerlig.



## Kontinuerlig

- Et **kontinuerlig** lydsignal er et lydsignal som funksjon av tid der signalstyrken kan ha uendelig høy presisjon, og signalet eksisterer for alle verdier av tiden innenfor et intervall i tid.
- En variabel  $t$  som er definert over et intervall  $a < t < b$  er **kontinuerlig** dersom vi innenfor intervallet har uendelig oppløsning, slik at  $t$  kan ha hvilken som helst verdi innenfor grensene.

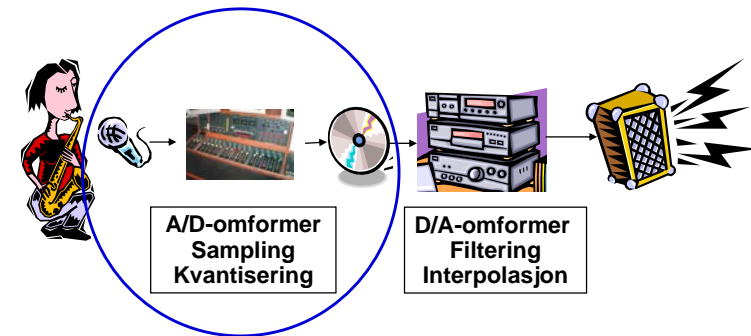
## Digital

- Begrepet **digital** brukes for å beskrive at informasjonen er representert ved diskrete signaler, eller ved at et bestemt signal finnes eller ikke finnes
- i bestemte posisjoner,
  - til bestemte tider,
  - ved bestemte frekvenser;
  - eller ved kombinasjoner av disse.

## Diskret

- Vi sier at en variabel  $t$  som er definert i et intervall  $a < t < b$  er **diskret** dersom vi har en **endelig** oppløsning innenfor intervallet, slik at settet med mulige verdier av  $t$  innenfor grensene er begrenset.
- En **diskret-tid** representasjon av et lydsignal er altså et lydsignal som bare er målt **på visse tidspunkter**.

## Fra artist via CD til ditt øre

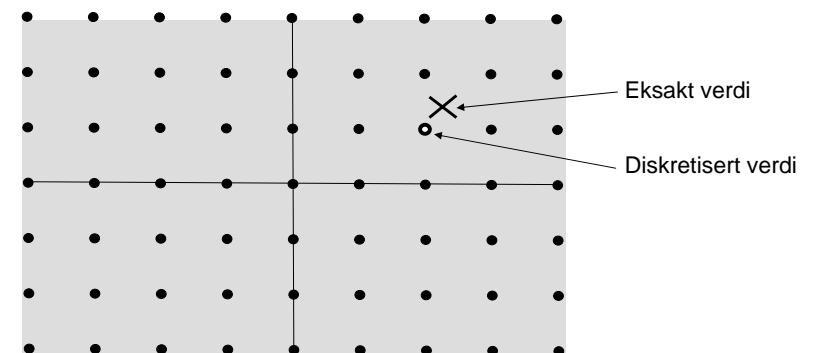


## Samplingsintervall og samplingsfrekvens

- Sampling er første skritt i en digitalisering.
- Tiden diskretiseres, og vi tar bare vare på en sekvens av lydintensitetsverdier  $y(t)$  ved bestemte diskrete tidspunkter.
- Sekvensen  $y[n]$ , for  $n = 0, 1, 2, 3, \dots, N-1$  er en diskret-tid (eller digital) representasjon av det kontinuerlige signalet  $y(t)$ .
- Tidsavstanden mellom to sampler, gitt i sekunder (eller millisekunder) kalles **samplingsintervallet**,  $T_s$
- Hvis samplingsintervallet er det samme mellom alle sampler sier vi at vi har en **uniform sampling**.
- Samplingsfrekvens** (også kalt **samplingrate**) angis i Hz, og er det inverse av samplingsintervallet,  $f_s = 1 / T_s$
- Samplingen gjøres av en **A/D-omformer** ved at lydintensiteten (i praksis mikrofonspenningen) holdes i et lite tidsintervall.

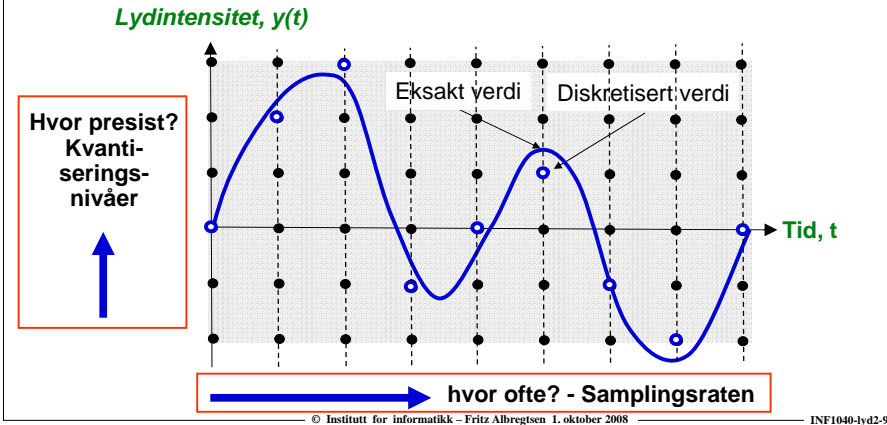
## Repetisjon: Diskretisering i rommet

- Punkter i flerdimensjonale rom må "snappes" til nærmeste representerbare punkt – på samme måte som tall (punkter i det endimensjonale rom) "snappes" til nærmeste representerbare tall



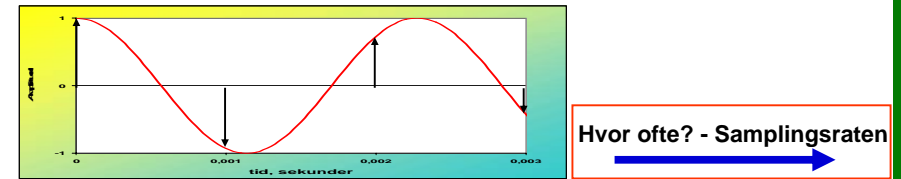
## Diskretisering av lyd

- Vi bestemmer på forhånd hvor ofte vi skal måle ("sample") lydintensiteten. Dette er *samplingsfrekvensen*.
- Selve målingen *kvantiseres* ("snappes") til nærmeste (eller nærmeste mindre) representerbare verdi.



## Diskret representasjon av et kontinuerlig lydsignal

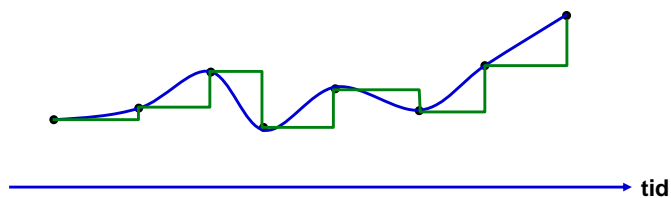
- Alle lydsignaler må samples og kvantiseres for å kunne lagres digitalt.



- Prosessen som gjør signalet *diskret i tid* kalles *sampling*.
- Å sample et signal vil si å plukke ut verdien til signalet på tidspunkter med gitte intervaller.
- Lydintensiteten ved tiden  $t$ ,  $y(t)$  må så *kvantiseres* for å kunne lagres (f.eks. til un-signed byte som gir verdiene 0-255).

## Lydintensiteten holdes fast i samplingsintervallet

- Den kontinuerlige kurven vi skal sample
- Tidspunktene vi skal sample på
- Sampling gjøres av A/D-omformeren ved at en lydintensitet / spenning holdes fast i et lite tidsintervall (samplingsintervallet)



## Kvantisering - et begrep

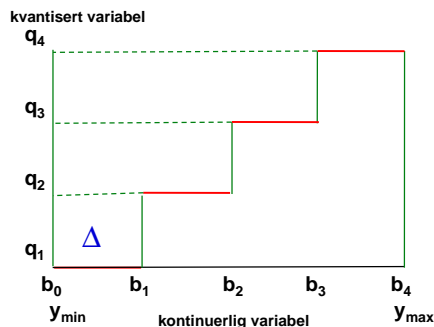
- fra latin (*quantitas*, mengde, antall; *quantus*, hvor stor).
- *kvantifisere*
  - om det å uttrykke noe relativt presist i en målbar størrelse, til forskjell fra en kvalitativ, omtrentlig angivelse.
- *kvantisere*
  - en prosess som avrunder signal-amplituden til alle leddene i en sekvens av sampler av et kontinuerlig signal  $y(t)$  til bestemte diskrete verdier – gjerne heltall.
- "*quantization*"
  - "the division of the amplitude range of a continuously variable signal into discrete levels for the purpose of sampling and coding"
- "*quantize*"
  - "to limit a variable to values that are integral multiples of a basic unit."

*Chambers Dictionary of Science and Technology*

*Collins English Dictionary*

## Uniform kvantisering

- Anta at spenningen varierer mellom  $y_{\min}$  og  $y_{\max}$ .
- Anta at vi deler området mellom  $y_{\min}$  og  $y_{\max}$  i  $N$  **kvantiseringsnivåer**.
- Da er  $B = \{b_0, b_1, \dots, b_N\}$  de  $N+1$  **beslutningsgrensene** som bestemmer hvilket nivå en målt spenning  $y(t)$  havner i.
- Vi har  $N$  forskjellige outputverdier eller **rekonstruksjonsnivåer**,  $Q = \{q_1, q_2, \dots, q_N\}$ .
- Hvis alle kvantiseringsnivåene har lik bredde,  $\Delta$ , kalles dette **uniform kvantisering**.
- Her illustrert for  $N = 4$  nivåer med lik bredde  $\Delta$ .

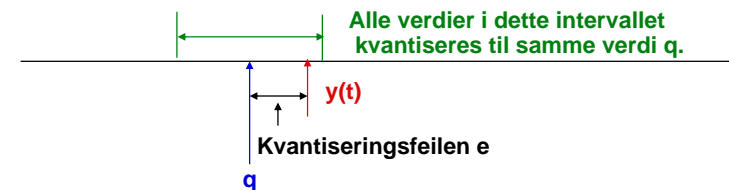


Hvor presist?  
Kvantiseringsnivåer



## Kvantiseringsfeil

- Anta at  $y(t)$  rekonstrueres til verdien  $q$ .
- Da får vi en kvantiseringsfeil  $e = y(t) - q$ .
- La oss si at vi velger  $q$  slik at den ligger midt i kvantiseringscellen ("midtredd", se lysark INF1040-lyd2-19).
- Den største kvantiseringsfeilen vi kan få for et sampel er da halvparten av bredden på en kvantiseringscelle:



## Signal/støyforhold

"Signal/støyforholdet" (Signal-to-Noise Ratio, SNR) uttrykkes ofte i desibel

- En høy SNR-verdi forteller at støyen er lite hørbar i forhold til signalet.
- Hvis vi uttrykker omfanget av signalet med  $y$  og omfanget av støyen med  $s$ , så kan signal/støy-forholdet uttrykkes ved

$$SNR = 10 \log_{10} \left( \frac{y^2}{s^2} \right) = 20 \log_{10} \left( \frac{y}{s} \right)$$

## Kvantiseringsstøy (for de interesserte)

- Kvantiseringsfeil kan gi opphav til hørbar kvantiseringsstøy.
- Anta at vi bruker  $N$  biter per sampel, og at det analoge signalet ligger i området fra  $-y_{\max}$  til  $+y_{\max}$ .
- Da har signalet en største verdi på omtrent  $2^{N-1}$ .
- Den største kvantiseringsfeilen er  $1/2$ , som vi har sett ovenfor.
- Vi får da for det såkalte signal/kvantiseringsstøy-forholdet ("Signal-to-Quantization Noise Ratio"):

$$SQNR(dB) = 20 \log_{10} \left( \frac{2^{N-1}}{1/2} \right) = 20 \log_{10} (2^N) = 20 N \log_{10} (2) \approx 6 N$$

- Hver ekstra bit vi bruker til et sampel gir oss ca 6 dB bedre SQNR.
- 16 biter (CD-standard) gir oss en SQNR på minst  $16 \times 6 \text{ dB} = 96 \text{ dB}$ .

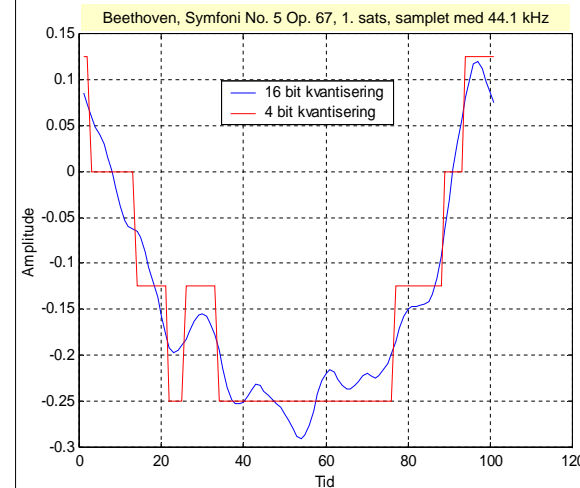
## Hvor mange kvantiseringsnivåer trenger vi?

- Når vi skal velge antall kvantiseringsnivåer, kan vi bruke det vi vet om hva øret kan oppfatte. Kvantiseringsfeil som er så små at vi ikke kan høre dem, er OK.
- Fordi vi stiller ulike krav til kvalitet, kan vi velge ulike nivåer for tale, musikk og telefoner.
  - Musikk-CD: 16 biter (65 536 nivåer)
  - Digital telefon: 8 biter (256 nivåer)
- Fordi kvantiseringsnivåene representeres i det binære tallsystemet, velger vi antall kvantiseringsnivåer som en potens av 2, altså  $2^n$ .
- $n$  er dermed antall biter per sampel (den såkalte *ordlengden*).

$$n = \log_2(\text{antall nivåer})$$



## Kvantisering av Beethovens 5. symfoni



16 biters kvantisering høres slik ut:



lyd36.wav

Er det lurt å komprimere lydfilen ved å redusere ordlengden fra 16 til 4 biter?

4 biters kvantisering høres slik ut:



lyd37.wav

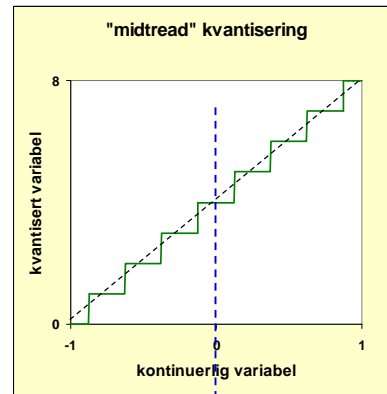
Altså: Ikke særlig lurt!

## "Midtread" kvantisering

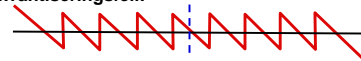
- Vi har så langt skalert og kvantisert en kontinuerlig variabel  $y(t)$  i området fra  $y_{\min}$  til  $y_{\max}$  til heltallene fra 0 til  $G-1$  ved å runde av til nærmeste heltall:

$$f = \text{Round} \left( G \frac{y(t) - y_{\min}}{y_{\max} - y_{\min}} \right)$$

- Dette kalles "midtread"-kvantisering.
- "Midtread"-kvantisering gir oss bl.a. en kvantiseringscelle sentrert omkring null, hvilket kan være en fordel hvis inputsignalet  $y(t)$  ofte er 0.
- Ulempe: Vi får  $G+1$  kvantiseringsnivåer, og første og siste kvantiseringscelle er ikke like brede som de andre.



kvantiseringsfeil:



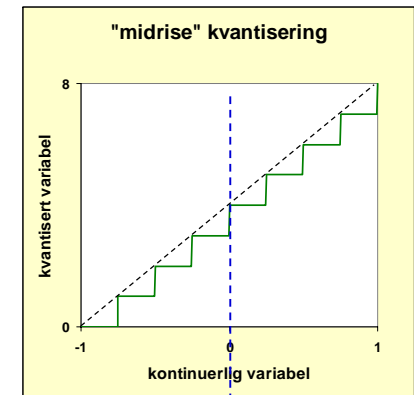
## "Midrise" kvantisering

- Et alternativ er "midrise"-kvantisering, der vi alltid går ned til største heltall mindre eller lik  $y(t)$  (floor-funksjonen).

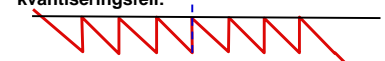
- Da blir ligningen

$$f = \text{Int} \left( G \frac{y(t) - y_{\min}}{y_{\max} - y_{\min}} \right)$$

- Hvis vi i tillegg klipper verdien som svarer til  $y_{\max}$ , får vi  $G$  output-verdier, og alle kvantiseringsceller er like brede.
- "Midrise"-kvantisering kan imidlertid gi kvantiseringsstøy hvis signalet  $y(t)$  er nær 0.



kvantiseringsfeil:



## Nyquist-teoremet

- Vi har et fundamentalt spørsmål:
- Hvor ofte må vi måle lydintensiteten i et gitt lydsignal?
- Harry Nyquist og Claude Shannon ga oss følgende teorem:

Anta at et analogt signal er *bånd-begrenset*, dvs. at det ikke har sinuskomponenter med frekvenser over et maksimum  $f_{max}$ . Signalet kan da rekonstrueres eksakt fra de samplene vi har, hvis samplingsfrekvensen  $f_s = 1/T_s$  er større enn  $2f_{max}$ .

- $2f_{max}$  kalles **Nyquist-raten**.
- Hvis vi sampler med en samplingsfrekvens  $f_s$  som er minst  $2f_{max}$ , så sampler vi i henhold til Nyquist-teoremet.
- Anta at vi skal sample lyd slik at vi kan rekonstruere alle hørbare frekvenser (opp til 20 000 Hz) uten noen feil eller forvrenging. Hva er  $f_s$ ?

## Sampling med Nyquist-raten

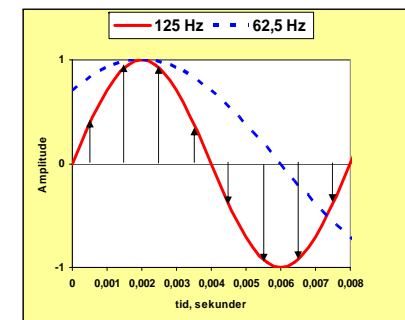
- Diskret til analog rekonstruksjon gir som resultat den kombinasjon av sinusoider som
  - - er kompatibel med det diskrete signalet
  - - har lavest mulige frekvenser
- Sampler vi med lavere frekvens enn Nyquist-raten og senere forsøker å rekonstruere det analoge signalet, får vi rekonstruert feil signal.
- I praksis oversampler vi med en viss faktor for å få god rekonstruksjon.

## Oversampling og undersampling

- **Oversampling**  
betyr at vi sampler med en høyere samplingsrate enn  $2f_{max}$ 
  - I rekonstruksjonen gjøres en filtrering/prosessering for å få tilbake det analoge signalet.
  - Denne filtreringen blir enklere og kan gjøres raskere hvis vi oversampler.
  - Vi vil også få redusert kvantiseringsstøyen.
  - Dersom vi oversampler med for høy faktor, øker lagringsbehovet unødvendig mye.
- **Undersampling**  
betyr at vi sampler med lavere samplingsrate enn  $2f_{max}$ 
  - Da får vi forvrenginger i frekvensinnholdet/aliasing.

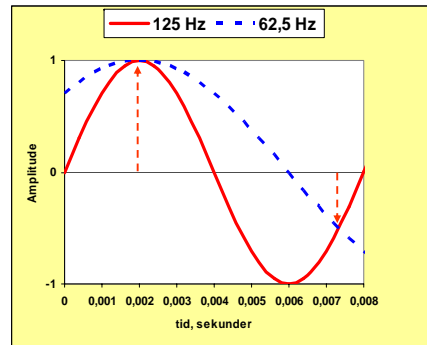
## Oversampling

- Vi sampler for eksempel en  $f = 125$  Hz sinus med samplingsfrekvens  $f_s = 1$  kHz som gir  $f_s/f = 1000/125 = 8$  sampler pr periode.
- Dette er en **4X oversampling** (svarte piler i figuren).



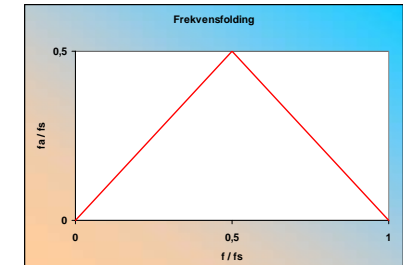
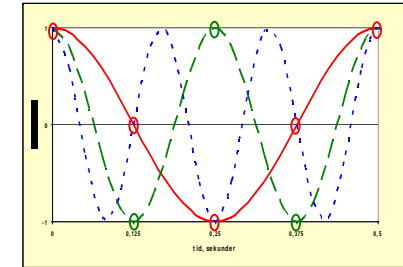
## Undersampling og aliasing

- Aliasing betegner det fenomenet at en sinusoid ved for lav samplingsrate gir opphav til samme diskrete signal som en sinusoid med lavere frekvens.
- Vi sampler for eksempel en  $f = 125$  Hz sinus med  $f_s = 1.5 f = 187.5$  Hz.
- Dette gir for eksempel sampler ved  $t = 0.002$  og ved  $t = 0.733$  (stiplede røde piler).
- Rekonstruksjon gir sinus med  $f_a = 62.5$  Hz, (stiplet kurve i figuren).
- Vi har fått en "aliasing".
- Merk at  $f_a = f_s - f$  når  $f < f_s < 2f$

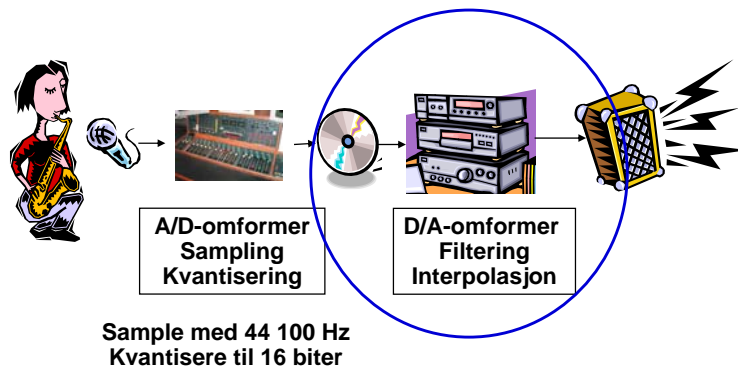


## Frekvensfolding (for de interesserte)

- I figuren til høyre vises  $\frac{1}{2}$  sekund av tre sinus-funksjoner med frekvenser 2, 4 og 6 Hz.
- Vi sampler med en fast  $f_s = 8$  Hz
- Vi får rekonstruert sinusoider med henholdsvis  $f = 2, 4$  og  $2$  Hz.
- $f$  som er under halvparten av  $f_s$  blir rekonstruert til korrekt frekvens
- $f$  mellom  $\frac{1}{2} f_s$  og  $f_s$ , blir rekonstruert til  $f_a = f_s - f$

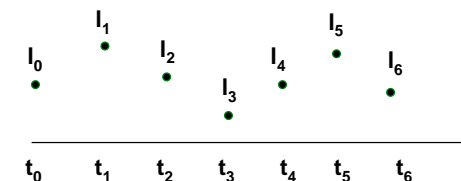


## Fra artist via CD til ditt øre



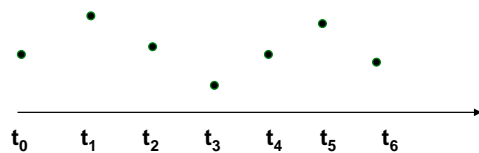
## Fra samplede punkter til avspilling

- Når lydsignalet skal spilles av, må det konverteres tilbake til et kontinuerlig / analogt signal igjen.
- Denne konverteringen kalles D/A-omforming (Digital -> Analog) og gjøres f.eks. i CD-spilleren (eller i programmet **play**).
- For å spille av lydsignalet, må vi vite samplingsraten.
  - Vi må vite tidsintervallet mellom et sampel og det neste.
- Avspillingsprogrammet vet at på tidspunkt  $t_1$  var lydstyrken  $I_1$ , og på tidspunkt  $t_2$  var lydstyrken  $I_2$ .



## Mer om avspilling

- ❑ Avspillingsprogrammet må også fylle ut med lydverdier mellom hvert sampel, dvs. f.eks. mellom tidspunkt  $t_0$  og tidspunkt  $t_1$ .
- ❑ Dette kan gjøres på flere måter (algoritmer), f.eks. ved å
  - Sette inn kopier av forrige sampel
  - Lineær interpolasjon
  - Høyere-ordens interpolasjon

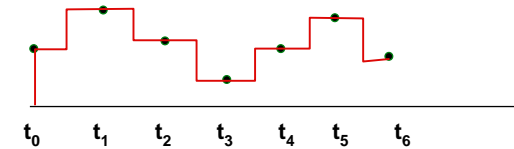


Jf. lærebokas avsnitt 8.4 om interpolasjon!



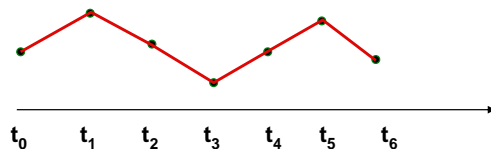
## Avspilling ved å sette inn kopier av sampel

- ❑ Denne algoritmen sier at lydsignalet har samme lydstyrke på alle tidspunkt mellom samplene, f.eks. mellom tidspunkt  $t_0$  og tidspunkt  $t_1$ .
- ❑ Dette gjøres vanligvis ved at vi skifter fra lydstyrke  $I_0$  til  $I_1$  på et tidspunkt midt mellom sampel  $t_0$  og  $t_1$ , osv.
- ❑ Med denne metoden skifter signalet brått fra en verdi til en annen. Brå skifter gir ikke alltid god lyd !
- ❑ Husk at for å representere slike brå skift trengs det mange frekvenskomponenter, helt opp til de mest høyfrekvente !



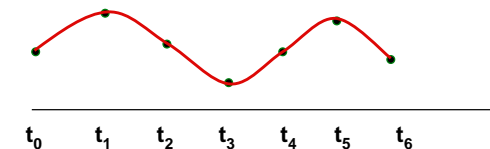
## Avspilling ved lineær interpolasjon

- ❑ Algoritmen sier at lydsignalet har endrer seg gradvis fra den lydstyrken det hadde på tidspunkt  $t_0$  til den lydstyrken det skal ha på tidspunkt  $t_1$ .
- ❑ Lydstyrken på tidspunkt mellom  $t_0$  til  $t_1$  finnes ved å legge en rett linje mellom verdien på tidspunkt  $t_0$  og verdien på tidspunkt  $t_1$ .
- ❑ Med denne måten blir det ikke så brå skifter i lydstyrke.
- ❑ Den avspilte lyden vil ikke inneholde så høye frekvenser som i forrige algoritme. (Husk lyden av trekantpulser !)



## Avspilling ved høyere-ordens interpolasjon

- ❑ Algoritmen antar at lydsignalet har endrer seg gradvis fra den lydstyrken det hadde på tidspunkt  $t_0$  til den lydstyrken det skal ha på tidspunkt  $t_1$ .
- ❑ Lydstyrken på tidspunkt mellom  $t_0$  til  $t_6$  finnes ved å tilpasse en glatt kurve (f.eks. et polynom) mellom verdiene på tidspunkt  $t_0$  og verdien på tidspunkt  $t_1$ , osv.
- ❑ Kurven skal gå gjennom alle punktene  $t_0, t_1, t_2, t_3, t_4, t_5$  og  $t_6$ , og den skal være glatt i skjøtene (for eksempel en spline-kurve).
- ❑ Med denne metoden blir det glatte overganger i lydstyrke .





## Kommentar til avspilling i praksis

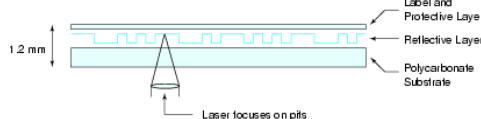
- I virkeligheten er det som skjer i en moderne CD-spiller en kombinasjon av både digital og analog signalbehandling.
- Lydsignalet oppsamples, og filtreres både med software (digitale filtere) og i elektronikken (analoge filtere).
- Når produsenter og selgere av CD-spillere reklamerer med for eksempel "8 X oversampling" ....
  - ...så betyr det ikke at de tryller fram 7 ekstra sampler av det originale signalet
  - ...det betyr bare at det legges inn 7 kopier av hvert sampel før filtrering og DA-konvertering for å få en enklere (og som regel litt bedre) rekonstruksjon av signalet.

## Lagring av digital lyd

- Anta at vi har samplet med samplingsraten  $f_s$ , og at vi har samplet minst i henhold til Nyquist raten.
- Hvis lydsignalet varer i N sekunder, må vi lagre  $N \cdot f_s$  sampler.
- Hvert sampel lagres med valgt datatype, vanligvis 16 bit integer.
- Hvis vi lagrer i stereo, lagrer vi 2 kanaler, hver med f.eks. 16 biter.
- Skal vi lagre 1 time musikk som er samplet med samplingsfrekvens 20 000Hz, blir lagringsbehovet
  - $20\,000 \cdot 3\,600 \cdot 16$  biter = 1 152 000 000 biter  
= 144 000 000 byte = 144 MB (288 MB i stereo).

## Hva ligger på en musikk-CD?

- Data lagres som bitmønstre i spesielle posisjonsceller/bins.



- Standard for CD audio kalles Red Book
- Audio-CD'er deles inn i følgende enheter:
  - Kapasiteten er 74 "minutter" a 60 "sekunder".
  - Enheten "sekund" har 75 "blokker"
  - En "blokk" består av 98 "rammer"
  - En "ramme" inneholder 24 byte data
- Avspilling skjer ved at rotasjonshastigheten varieres mens platen avspilles (innenfra og utover) slik at 75 blokker leses pr. sekund med datarate:
  - $R = 44\,100 \text{ sampler/s} \cdot 2 \text{ byte} \cdot 2 \text{ kanaler} = 176\,400 \text{ byte/s}$ .

## Redundans og maskering av lyd

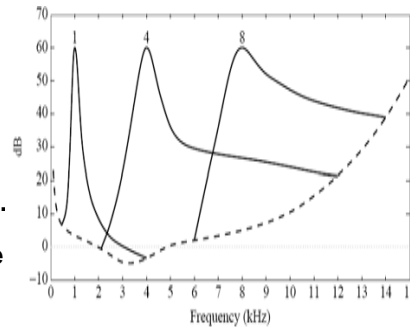
- Det er en god del lyd vi faktisk ikke kan høre på grunn av annen lyd, og som vi derfor heller ikke nødvendigvis trenger å lagre.
- Dette skyldes flere effekter:
  - Frekvensmaskering
  - Tidsmaskering
  - Stereoredundans

*Dette er grunnlaget for avansert komprimering, for eksempel i MP3-lydformatet*



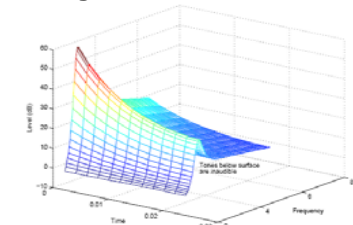
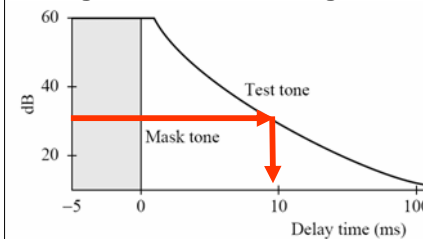
## Frekvensmaskering

- ❑ For en gitt frekvens finnes et "kritisk frekvensbånd" der denne lyden vil maskere ut andre lyder. Figuren viser maskeringsterskelen for tre toner (lydintensitet 60 dB).
- ❑ Noen hovedpunkter :
  - Frekvensmaskeringen er asymmetrisk.
  - Jo høyere lydintensitet, dess bredere frekvensmaskering.
  - Frekvensmaskeringen er bredere for høyere frekvenser.
- ❑ Vi filtrerer lyden til separate signaler på 32 frekvensbånd, istedenfor å måtte ta vare på signalstyrken på alle frekvenser.



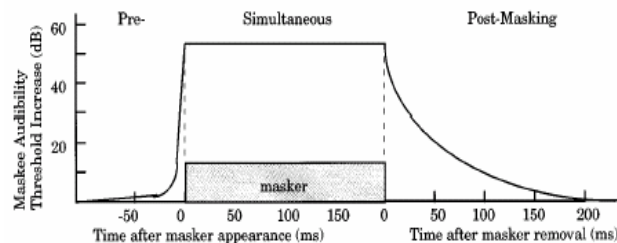
## Tidsmaskering

- ❑ Man kan spille en 1 kHz maskeringstone med lydintensitet 60 dB – og en nærliggende testtone, for eksempel 1.1 kHz med 30 dB.
- ❑ Pga frekvensmaskering kan ikke testtonen høres.
- ❑ Når maskeringstonen slås av kan vi høre den - men først etter at det har gått litt tid.
- ❑ Vi kan finne maskeringstiden for flere lydintensiteter og frekvenser, og finne 2D maskeringsterskel i frekvens og tid.



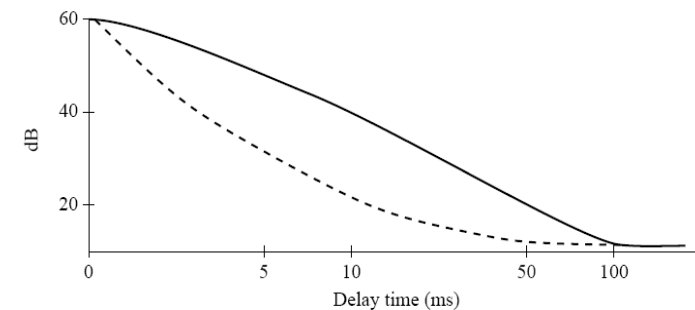
## Postmaskering og premaskering

- ❑ Tidsmaskering skyldes at hårcellene i det indre øre går i metning, og trenger litt tid før de igjen reagerer normalt.
  - "Postmaskering" har et effektivt tidsintervall på 50-200 ms.
- ❑ En sterk lyd kan maskere en svakere lyd som spilles like før
  - "Premaskering" har mye kortere effektivt tidsintervall, 2-5 ms.



## Metning avhenger av varighet ...

- ❑ Metningsfenomenet avhenger av hvor lenge maskeringstonen har blitt spilt.
    - Stiplet kurve viser maskeringstid når tonen er spilt i 100 ms
    - Heltrukket kurve når tonen er spilt i 200 millisekunder.
- (Fra Li and Drew, Fundamentals of Multimedia, 2004).



## Stereoredundans

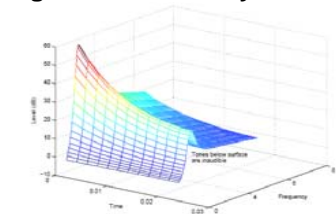
- ❑ To kanaler gir et mye rikere lydbilde enn bare en kanal.
- ❑ I mange tidsintervaller er det vanligvis stor likhet mellom signalene i de to kanalene.
- ❑ Vi kan beregne :
  - Korrelasjonen mellom kanalene
  - Differansen mellom kanalene
- ❑ Det kan være mest effektivt å lagre
  - det ene signalet...
  - ... pluss differansen mellom dem.
- ❑ Da kan vi alltid komme tilbake til de to originale signalene.

## Maskering av lyd

- ❑ Maskering: “Én lyd er ikke hørbar samtidig med en annen lyd”

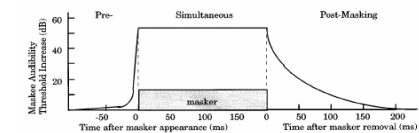
### 1. Simultan maskering

- Støy kan maskere en tone
- En tone kan maskere støy
- Støy kan maskere støy

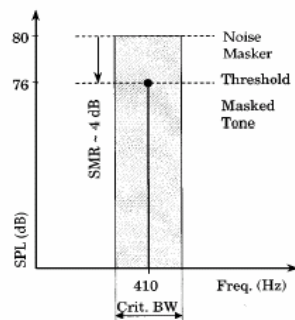


### 2. Tidsmaskering

- Premaskering (ca 2 ms)
- Postmaskering (ca 100 ms)



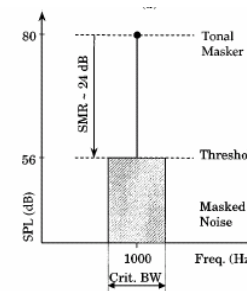
## Eksempel 1: Støy kan maskere en tone



Vi kan ikke høre en sinusoid som ligger i samme kritiske bånd som støyen hvis lydtryknivået er under en viss terskel.

Filtrert støy	Tone 1	Tone 2	Støy + Tone 1	Støy + Tone 2
Senter 410 Hz Bredde 111 Hz	820 Hz 5 dB under støyen	410 Hz 5 dB under støyen		
			Ikke maskert	Maskert

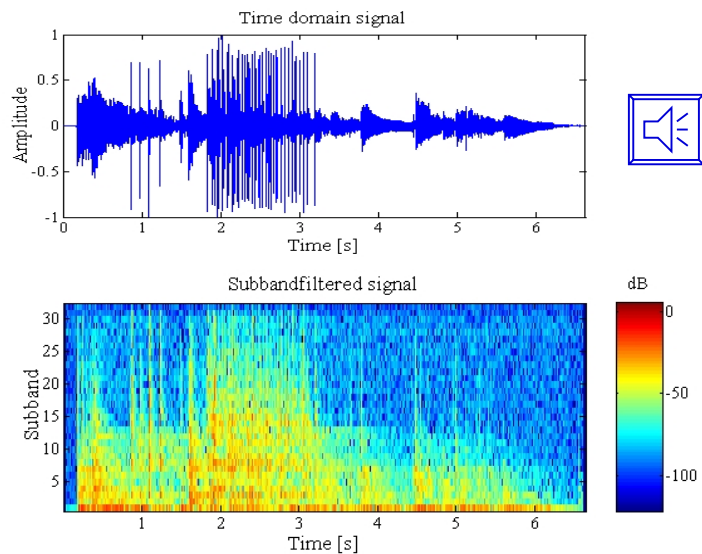
## Eksempel 2: En tone kan maskere støy



Vi kan ikke høre en filtrert støy som ligger i det samme kritiske bånd som en sinusoid, hvis lydtrykket er under en viss terskel.

Filtrert støy	Tone 1	Tone 2	Støy + Tone 1	Støy + Tone 2
Senter 1 kHz Bredde 162 Hz - 15 dB	2 kHz	1 kHz		
			Ikke maskert	Maskert

## Lydeksempel: Kastanjetter og gitar



INF1040-lyd2-45

## Eksempler på kompresjon av lyd



Kompresjonsfaktor	2	4	8
Direkte kvantisering - kvantiseringsfeil (SQNR)	8 biter	4 biter	2 biter
Nedsampling (halv samplingsfrekvens) og kvantisering	16 biter	8 biter	4 biter
MP1 - kompresjonsfeil (SQNR)		4biter	2 biter

© Institutt for informatikk – Fritz Albrechtsen 1. oktober 2008

INF1040-lyd2-46

## Digitalisering av lyd – oppsummering

- ❑ Et kontinuerlig lydsignal er en reell funksjon  $y(t)$  av tiden  $t$ .
- ❑ Digitaliseringsprosessen består av tre enkle steg:
  1. Bestem samplingsintervallet  $T_s$  (bruk Nyquist-kriteriet!) og ta  $N$  diskrete sampler  $y[n]$ ,  $n=0,1,2,3, \dots, N-1$ .
  2. Skalér verdien av alle samplene, for eksempel til verdiområdet 0–255 (8 biter).
  3. Kvantiser verdiene av de skalerte samplene ved å avrunde til nærmeste heltallsverdi (eller gå ned til nærmeste heltallsverdi).
- ❑ Legg merke til at vi ikke trenger å ta vare på verdien av tiden eller verdien av  $n$  for hvert sampel.  
Så lenge vi lagrer dem etter hverandre og kjenner  $T_s$ , er tidspunktet for hvert sampel implisitt gitt.
- ❑ Vi kan spare plass ved å la være å lagre det vi ikke hører.

© Institutt for informatikk – Fritz Albrechtsen 1. oktober 2008

INF1040-lyd2-47