

LYDBØLGER

Mål: Hensikten med oppgaven er å gjøre seg kjent med bølger og den matematiske beskrivelsen av bølger. Som del av dette inngår å gjøre seg kjent med begrepene bølgelengde, bølgetall, frekvens, vinkel-frekvens (sirkelfrekvens) og fasehastighet. Det legges vekt på å knytte sammen den matematiske beskrivelsen og den fysiske forståelsen av bølgefenomener. I oppgaven vil en møte tre ulike varianter av bølger; løpende bølger, stående bølger og resonansbølger. Stående bølger oppstår når en løpende og reflektert bølge adderes i området foran reflektoren. Når mediet som bølgene forplanter seg i reflekteres fra to ulike grenseflater, kan resonans oppstå. Et eksempel på dette er et rør som enten er åpent eller lukket i endeflatene. Da vil frekvenser som samsvarer med lengden av luftsynderen forsterkes.

I det som følger nedenfor vil først oppgavene og noen kommentarer i forbindelse med gjennomføringen bli introdusert. Som støtte for eksperimentene introduseres bølgelikningen for endimensjonale bølger med løsninger i teoridelen. Dernest vil det bli gitt en kort beskrivelse av løpende og stående bølger, og hvorfor stående bølger og resonans oppstår. En kortfattet versjon av Fourieromvending blir også presentert.

Oppgaver

- 1: Finn fasehastigheten til lydølger ved bruk av pulsed eksitasjon av luftøylen i et rør (Kundts rør).
- 2: Finn fasehastigheten til lydølger ved å analysere lydtrykket som funksjon av sted inne i et Kundts rør.
- 3: Bestem fasehastigheten til lydølger ved eksitasjon av Kundts rør med støy.
- 4: Bestem resonansfrekvensene til orgelpiper som eksiteres med støy.

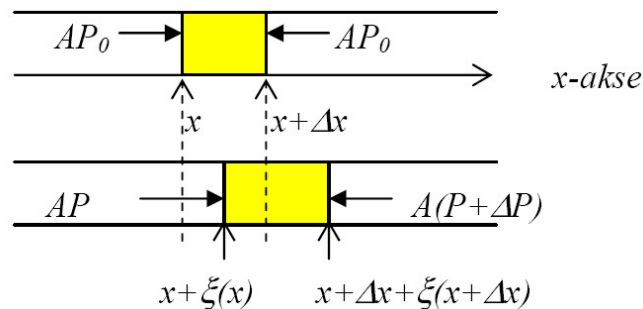
1 Teoretisk bakgrunn

1.1 Lydhastigheten

Lydølger skyldes forstyrrelser av lufttetthet og forplantning av disse trykkvariasjonene. For å utlede bølgelikningen for trykkølger brukes Newtons II lov på et element av lufta og adiabatlikningen for volumendringer av gasser.

Et luftvolum avgrenses, se Fig. 1, og P_0 er lufttrykket i likevektstilstanden. Avgrensingen av luftvolumet gjøres ved posisjon x og $x + \Delta x$ langs x -aksen. Δx er altså tykkelsen av luftvolumet og A er tverrsnittet til luftøylen. Vi tenker oss så at luftøylen forskyves, på grunn av trykkendringer, og at trykket på venstre og høyre grenseflate av luftøylen er P og $P + \Delta P$, henholdsvis. De nye endeflatene til luftøylen ligger ved $x + \xi(x)$ og $x + \Delta x + \xi(x + \Delta x)$, der $\xi(x)$ og $\xi(x + \Delta x)$ er endringene i posisjonene til endeflatene av luftmassen. Det akustiske trykket er trykkforskjellen $p = P - P_0$; eller $P = P_0 + p$. Dette medfører at endringer i trykket P er lik en endring i det akustiske trykket; $\Delta P = \Delta p$, siden P_0 er konstant.

Når vi bruker Newtons II lov (summen av krefter er lik masse multiplisert med akselerasjon) på dette



Figur 1: En avgrenset luftsøyle som forskyves på grunn av trykkendringer.

luftelementet, får vi:

$$P \cdot A - (P + \Delta P) \cdot A = A \cdot \Delta x \cdot \rho \cdot \frac{\partial^2 \xi}{\partial t^2} \quad (\text{Newton II}). \quad (1)$$

Symbolet ∂ betyr partiellderivert, altså en endring i størrelsen selv og ρ er tettheten til lufta. Herav følger:

$$-\Delta p = \Delta x \cdot \rho \cdot \frac{\partial^2 \xi}{\partial t^2} \quad (2)$$

$$-\frac{\Delta p}{\Delta x} \approx -\frac{\partial p}{\partial x} = \rho \cdot \frac{\partial^2 \xi}{\partial t^2}. \quad (3)$$

Når en avgrenset luftmengde presses sammen, vil trykket som lufta utsettes for og volumendringene være proporsjonale:

$$p = -B \cdot \frac{\Delta V}{V} \quad (4)$$

der B kalles for volumkompressibiliteten. Dette gir:

$$p = -B \cdot \frac{A \cdot \partial \xi}{A \cdot \partial x} = -B \cdot \frac{\partial \xi}{\partial x}. \quad (5)$$

Når dette deriveres med hensyn på x , fås:

$$\frac{\partial p}{\partial x} = -B \cdot \frac{\partial^2 \xi}{\partial x^2} \quad (6)$$

og når dette settes inn i likningen over, fås bølgelikningen:

$$B \frac{\partial^2 \xi}{\partial x^2} = \rho \cdot \frac{\partial^2 \xi}{\partial t^2} \quad (7)$$

$$\frac{\partial^2 \xi}{\partial x^2} = \frac{\rho}{B} \cdot \frac{\partial^2 \xi}{\partial t^2} = \frac{1}{c^2} \cdot \frac{\partial^2 \xi}{\partial t^2} \quad (8)$$

der $c^2 = \frac{B}{\rho}$ og c er fasehastigheten til bølgene.

Vi vil nå finne et uttrykk for volumkompressibiliteten. Endringene i luftvolumet er så raske at de regnes som adiabatisk, som vil si at ikke varme fra omgivelsene rekker å slippe ut fra den komprimerte lufta. Da gjelder følgende sammenheng mellom trykk og volum:

$$pV^\gamma = \text{konstant}, \quad (9)$$

der γ er adiabatkonstanten.

Derivasjon av dette uttrykket gir, når høyresiden er konstant:

$$dp \cdot V^\gamma + p \cdot \gamma \cdot V^{\gamma-1} \cdot dV = 0 \quad (10)$$

eller

$$dp + p \cdot \gamma \cdot \frac{dV}{V} = 0 \quad (11)$$

som gir

$$\frac{dp}{dV} = -p \cdot \gamma \quad (12)$$

som er B med motsatt fortegn. Tettheten til lufta er lik massen til lufta, som er antall mol (n) multiplisert

med molekylvekta M (molmassen, g/mol) til luft, dividert på volumet denne massen opptar: $\rho = \frac{m}{V} = \frac{n \cdot M}{V}$. Dermed blir fasehastigheten for lydbølger:

$$c = \sqrt{\frac{B}{\rho}} = \sqrt{\frac{p\gamma}{\rho}} = \sqrt{\frac{\frac{nRT}{V} \cdot \gamma}{\frac{n \cdot M}{V}}} = \sqrt{\frac{\gamma RT}{M}} \quad (\text{lydhastigheten}). \quad (13)$$

Dimensjonen (enheten) av uttrykket under rottegnet blir:

$$\left[\frac{\gamma RT}{M} \right] = \frac{1 \cdot \text{Nm}/(\text{mol} \cdot \text{K}) \cdot \text{K}}{0,001 \text{ kg/mol}} = \frac{1000 \cdot \text{Nm}}{\text{kg}} = \frac{\text{m}^2}{\text{s}^2}. \quad (14)$$

Molekylmassen M er antallet gram av molekylvekta (tilnærmet lik antallet nukleoner i molekylet). Tallverdien til molekylmassen til luft, som er en blanding av nitrogen og oksygen, er; $M = 28,8$ g (fra $0,8 \cdot 2 \cdot 14 + 0,2 \cdot 2 \cdot 16 = 28,8$) slik at lydhastigheten som funksjon av temperatur T (i Kelvin) blir:

$$c = \sqrt{\frac{1,4 \cdot 8,314 \cdot 1000 \cdot T/\text{K}}{28,8}} \text{ m/s} = 20,05 \cdot \sqrt{T/\text{K}} \text{ m/s}. \quad (15)$$

For null grader Celsius (273,14 K) blir lydhastigheten: $c = 331,4$ m/s. Dersom en vil finne lydfarten som funksjon av temperaturen T_C i Celsiusgrader, kan en bruke relasjonen:

$$T = (273 + T_C/^\circ\text{C}) \text{ K} \quad (16)$$

og får da ved romtemperatur $T_C = 20^\circ\text{C}$ at $c = 343,4$ m/s.

Bølgelikningen er en partiell differensiallikning og er tilfredsstilt av alle to ganger deriverbare funksjoner av form $\xi = f(u) = f(x - ct)$. Dette ser en ved derivasjon og innsetting i bølgelikningen. Partiell derivasjon med hensyn til x gir:

$$\frac{\partial^2 \xi}{\partial x^2} = \frac{\partial^2 f}{\partial u^2} \cdot \frac{\partial u^2}{\partial x^2} = \frac{\partial^2 f}{\partial u^2} \cdot 1 \quad (17)$$

og partiell derivasjon med hensyn på t gir:

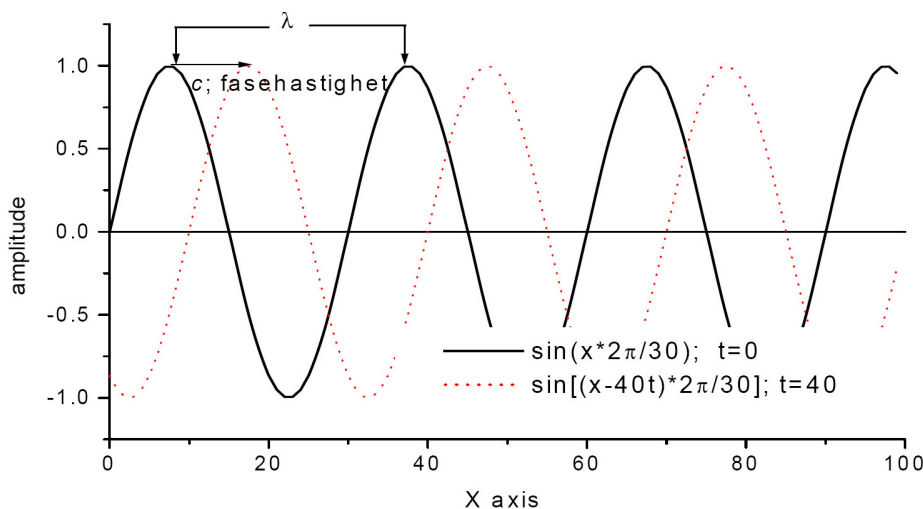
$$\frac{\partial^2 \xi}{\partial t^2} = \frac{\partial^2 f}{\partial u^2} \cdot \frac{\partial u^2}{\partial t^2} = \frac{\partial^2 f}{\partial u^2} \cdot c^2. \quad (18)$$

1.2 Løpende bølger (vandrebølger)

En harmonisk bølge har formen:

$$\xi(x, t) = \xi_0 \cdot \sin(\varphi) \quad (19)$$

der φ kalles fasen. Ved tidspunktet $t = 0$ og ved en vilkårlig posisjon x er fasen $\varphi = \frac{x}{\lambda} \cdot 2\pi$, der λ er bølgelengden eller avstanden mellom to bølgetopper. Når $x = \lambda$ vil sinusfunksjonen få samme verdi som i $x = 0$. x/λ er lik antallet bølger mellom 0 og x . Ved en vilkårlig tid t senere er fasen på stedet x lik $\varphi = \frac{x - ct}{\lambda} \cdot 2\pi$, siden bølgen da har beveget seg et stykke ct , se Fig. 2.



Figur 2: Vandrebølge.

Derfor skriver vi en harmonisk bølge på følgende form:

$$\xi(x, t) = \xi_0 \cdot \sin\left(\frac{2\pi}{\lambda} \cdot x - \frac{2\pi \cdot c \cdot t}{\lambda}\right) = \xi_0 \cdot \sin\left(\frac{2\pi}{\lambda} \cdot x - 2\pi f \cdot t\right) = \xi_0 \cdot \sin(kx - \omega t) \quad (20)$$

siden $\frac{c}{\lambda} = f$. Størrelsen k kalles bølgetallet og er antallet bølger på en strekning 2π , dvs. $k = 2\pi/\lambda$ og vinkelfrekvensen er $\omega = 2\pi \cdot f$.

$\xi(x, t)$ er utslaget til luftmolekylene fra likevekt ved posisjon x og tid t . Trykkvariasjonene kan finnes ved derivasjon:

$$p = -B \cdot \frac{\partial \xi(x, t)}{\partial x} = -B \cdot \xi_0 \cdot k \cdot \cos(kx - \omega t) \equiv p_0 \cdot \cos(kx - \omega t) \quad (21)$$

og det er oftest trykkendringer som måles.

Vi vil nå finne et uttrykk for akustisk impedans (Z), som er mediets motstand mot lyd. Den er definert som forholdet mellom akustisk trykk (p) og hastigheten (v) til luftmolekylene:

$$v = \frac{\partial \xi}{\partial t} = \xi_0 \cdot \cos(kx - \omega t) \cdot (-\omega) \quad (22)$$

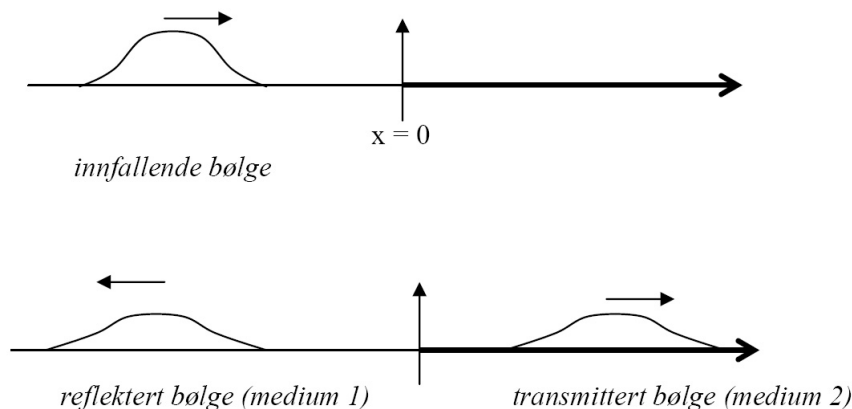
som gir:

$$Z = \frac{p}{v} = \frac{B \cdot k}{\omega} = \frac{B}{w} \cdot \frac{w}{c} = \frac{c^2 \cdot \rho}{c} = c \cdot \rho. \quad (23)$$

Den akustiske impedansen, som er analog til elektrisk motstand (spenning/strøm), er altså produktet av fasehastighet og mediets tetthet.

1.3 Refleksjon av bølger

Vi tenker oss en bølge som løper mot høyre med fasehastighet c . I $x = 0$ møter bølgen en forandring i mediet, for eksempel at tettheten endrer seg.



Figur 3: Innfallende, reflektert og transmittert bølge.

Da vil noe av bølgen forsette inn i det nye mediet, mens en del vil reflekteres tilbake, se Fig. 3. I tilfellet av harmoniske bølger beskrives den innfallende, reflekterte og den transmitterte bølgen beskrives slik:

$$\xi^i(x, t) = \xi_0^i \cdot \sin(kx - \omega t) \quad \xi^r(x, t) = \xi_0^r \cdot \sin(kx + \omega t) \quad \xi^t(x, t) = \xi_0^t \cdot \sin(k_t x - \omega t), \quad (24)$$

der symbolbruken er som tidligere. Når vi nærmer oss $x = 0$ fra venstre side, hvor bølgen er en sum av innfallende og reflektert bølge, eller høyre, der vi bare har den transmitterte bølgen, må utslagene være like:

$$\xi_0^i \cdot \sin(-\omega t) + \xi_0^r \cdot \sin(\omega t) = \xi_0^t \cdot \sin(-\omega t) \quad (25)$$

eller

$$\xi_0^i - \xi_0^r = \xi_0^t \quad (\text{første grensebetingelse}). \quad (26)$$

Videre må trykket være kontinuerlig i punktet $x = 0$. Dette medfører:

$$B_1 \cdot k_i \cdot \xi_0^i + B_1 \cdot k_i \cdot \xi_0^r = B_2 \cdot k_t \cdot \xi_0^t \quad (\text{andre grensebetingelse}). \quad (27)$$

Av dette finner vi amplituden til den reflekterte bølgen, når vi bruker at:

$$B \cdot k = \rho \cdot c^2 \cdot \frac{\omega}{c} = \rho \cdot c \cdot \omega = Z \cdot \omega \quad (28)$$

og at vinkelfrekvensen (ω) er like stor på begge sider:

$$\xi_0^r = \frac{Z_2 - Z_1}{Z_2 + Z_1} \cdot \xi_0^i \quad (\text{amplitude til reflektert bølge}). \quad (29)$$

En ser at når tettheten til mediet på høyre side er mye større enn på venstre side, vil amplituden til reflektert bølge være like stor som den innfallende.

1.4 Stående bølger

Når hele bølgen blir reflektert, blir bølgen en sum av innfallende og reflektert bølge:

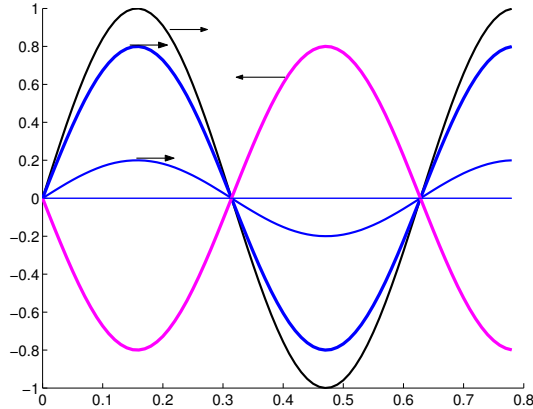
$$\begin{aligned} \xi(x, t) &= \xi_0^i \cdot \sin(kx - \omega t) + \xi_0^i \cdot \sin(kx + \omega t) \\ &= \xi_0^i \cdot [\sin kx \cdot \cos \omega t - \cos kx \cdot \sin \omega t] + \xi_0^i \cdot [\sin kx \cdot \cos \omega t + \cos kx \cdot \sin \omega t] \\ &= 2 \cdot \xi_0^i \cdot \sin kx \cdot \cos \omega t \quad (\text{stående bølge}). \end{aligned} \quad (30)$$

Dette er en harmonisk svingning ($\cos \omega t$) med en x -avhengig amplitude ($2\xi_0^i \cdot \sin kx$) og kalles en stående bølge.

Når amplituden til den reflekterte bølgen er mindre enn til den innfallende, blir bølgen en sum av en stående og en løpende bølge.

$$\xi(x, t) = 2\xi_0^r \cdot \sin kx \cdot \cos \omega t + (\xi_0^i - \xi_0^r) \cdot \sin(kx - \omega t). \quad (31)$$

Dette er også vist i Fig. 4



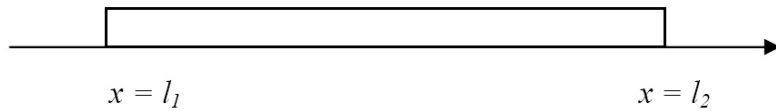
Figur 4: Den innfallende bølgen (svart) kan ses på som en superposisjon av de to blå bølgene. Den (rosa) reflekterte bølgen (her omtrent 75 % av den innfallende) interfererer med den største delen (tykke blå) av den innfallende bølgen, noe som resulterer i en stående bølge. Den tynne blå linjen representerer den delen som er den løpende bølgen. Figuren viser den løpende bølgen når dens noder tilfeldigvis samsvarer med de fra den stående. I laboratoriet måler man amplituden som en funksjon av tiden, og derfor vil den løpende bølgen alltid gi det samme bidraget til amplitude, det vil si et konstant offset.

1.5 Resonans

Begge ender er lukket:

Vi tenker oss at lydbølgene hindres i videre forplantning på to steder, slik at de reflekteres tilbake. Dette er tilfellet for bølgeutbredelse i et endimensjonalt rør. Anta at dette røret har lengde L og at den ene

enden er plassert i $x = l_1$ og den andre i $x = l_2$.



Figur 5: Rør med to lukkede ender.

Randbetingelsene blir da at utslagene (ξ) må være null både i $x = l_1$ og $x = l_2$. Det medfører, ifølge uttrykket for stående bølger; at $\sin kl_1 = 0$ og $\sin kl_2 = 0$. Dette medfører at både $kl_1 = n_1\pi$ og $kl_2 = n_2\pi$, der både n_1 og n_2 er hele tall. Dermed blir:

$$L = l_2 - l_1 = (n_2 - n_1) \frac{\pi}{\frac{2\pi}{\lambda}} = \frac{n \cdot \lambda}{2} \quad (32)$$

der n er et helt tall, $n = 1, 2, 3, 4, \dots$. Dette kan også skrives om til $\lambda_n = \frac{2 \cdot L}{n}$.

Begge ender er åpne:

Det er lettere for lydbølgen å forplante seg i fri luft sammenliknet med i det begrensede området inne i røret. Det medfører at en del av bølgen blir reflektert når den kommer til utgangen av røret. I åpningen av røret er trykket lik lufttrykket, slik at betingelsene blir da at det akustiske trykket er null $\cos(kl_1) = 0$ og $\cos(kl_2) = 0$, der l_1 og l_2 er posisjonene til endene av røret. Disse betingelsene gir $k \cdot l_1 = \frac{\pi}{2} + n_1 \cdot \pi$ og $k \cdot l_2 = \frac{\pi}{2} + n_2 \cdot \pi$ som til sammen gir:

$$L = l_2 - l_1 = \frac{(n_2 - n_1) \cdot \pi}{k} = \frac{n}{2} \cdot \lambda \quad (33)$$

altså $\lambda_n = \frac{2 \cdot L}{n}$ som er det samme som over.

En ende åpen, den andre lukket

Anta at åpningen til venstre er åpen og den til høyre er lukket. Da blir betingelsene: $\cos(kl_1) = 0$ og $\sin(kl_2) = 0$ som gir $k \cdot l_1 = \frac{\pi}{2} + n_1 \cdot \pi$ og $k \cdot l_2 = n_2 \cdot \pi$, henholdsvis.

Av dette får vi:

$$L = l_2 - l_1 = \frac{(n_2 - n_1) \cdot \pi}{k} + \frac{\pi}{2} \cdot \frac{1}{k} \quad (34)$$

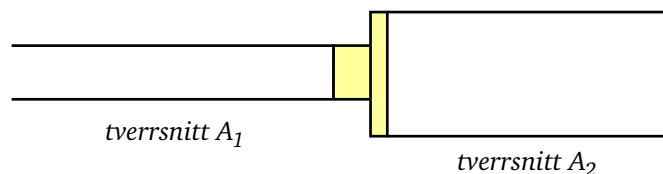
som gir:

$$\lambda_n = \frac{2 \cdot L}{n - \frac{1}{2}} \quad (35)$$

Som vi skal se senere, er disse uttrykkene en overforenkling.

1.6 Impedans og refleksjon i rør

Vi har fra tidligere at impedansen for plane lydbølger, $Z = p/v = c\rho$, er forholdet mellom lydtrykk og hastighet til luftmolekylene, og den kalles karakteristisk impedans. En stor impedans vil si at for et gitt lydtrykk vil molekylene bevege seg lite. Vi betrakter nå lydbølger inne i et rør sammensatt av to deler, med tverrsnittsareal A_1 og A_2 , henholdsvis.



Figur 6: Diskontinuitet i rør: ulike tverrsnitt.

Da må vi regne med impedans multiplisert med arealet. Dette kan fås tilbake til grensebetingelsen om at utslagene på grensa må være like. Når arealene er forskjellige, blir denne grensebetingelsen slik:

$$A_1 \cdot \xi_1 = A_2 \cdot \xi_2.$$

Dette uttrykker at endringer i luftvolum på begge av grensen er like store. Dermed blir forholdet mellom amplitudene til innfallende og reflektert bølge:

$$\frac{\xi_0^r}{\xi_0^i} = \frac{ZA_2 - ZA_1}{ZA_2 + ZA_1} = \frac{A_2 - A_1}{A_2 + A_1}. \quad (36)$$

I et rør med åpen ende vil det altså skje refleksjon og transmisjon av lydbølger.

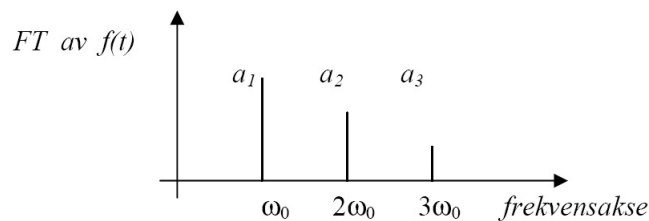
1.7 Litt om Fourieromvending

La oss tenke oss et signal som varierer med tiden på følgende måte:

$$f(t) = a_1 \cdot \sin(\omega_0 t) + a_2 \cdot \sin(2\omega_0 t) + \dots a_n \cdot \sin(n\omega_0 t) + \dots \quad (37)$$

Signalet er en sum av harmoniske signaler med frekvenser som er heltallige multipla av grunnfrekvensen ω_0 . Dette kunne være lydbildet fra f. eks. et instrument der a_1 er amplituden til grunnsvingningen ω_0 og a_n er amplituden til de overharmoniske svingningene med frekvens $n\omega_0$. Framstilt som funksjon av tiden t , eller i tidsbildet, er denne lyden en sum av harmoniske svingninger.

Enhver funksjon kan løses opp i en sum av harmoniske funksjoner, som kalles en Fourieromvending av signalet. I vårt eksempel er funksjonen allerede en sum av harmoniske signaler. Da kan amplitudene (Fourierkoeffisientene) framstilles som funksjon av frekvensene. I vårt eksempel får en bare bidrag ved ω_0 , grunnfrekvensen, og ved $n\omega_0$, de overharmoniske frekvensene, slik at den Fourieromvendte funksjonen kan vises umiddelbart; se Fig. 7.



Figur 7: Fourieromvending av signal som sum av harmoniske signaler.

Det generelle uttrykket for en Fourierrekka til periodisk funksjon $f(t) = f(t + T)$, der T er perioden, er:

$$f(t) = a_0 + \sum_{n=1}^{\infty} a_n \cdot \cos\left(\frac{2\pi n t}{T}\right) + \sum_{n=1}^{\infty} b_n \cdot \sin\left(\frac{2\pi n t}{T}\right), \quad (38)$$

der de såkalte Fourierkoeffisienten a_n og b_n er:

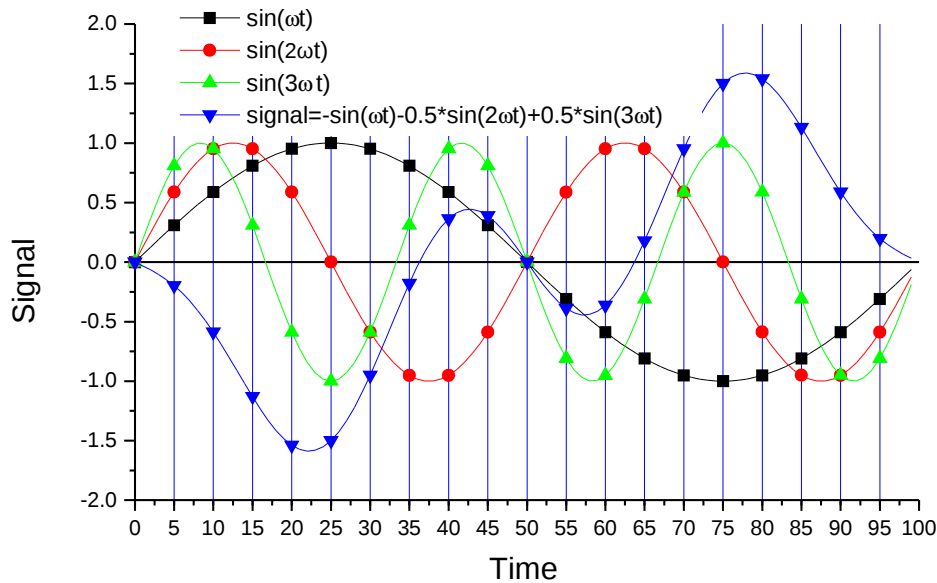
$$a_n = \frac{1}{\pi} \int_0^{2\pi} f(\varphi) \cdot \cos(n\varphi) d\varphi \quad b_n = \frac{1}{\pi} \int_0^{2\pi} f(\varphi) \cdot \sin(n\varphi) d\varphi \quad (39)$$

hvor $\varphi = \frac{2\pi}{T} \cdot t = \omega_0 t$.

Fourierkoeffisientene finnes altså ved en matematisk utregning av disse integralene. Dette er som regel vanskelig, og ofte gjøres dette i praksis hjelp av datamaskin. Utgangspunktet er da et signal (f) som omsettes til digital form med jevne mellomrom, og disse tidspunktene kan kalles t_k , der tallet $k = 0, 1, 2, 3, \dots, (N - 1)$. Vi kjenner altså funksjonsverdiene $f(t_k)$. En diskret Fourieromvending (DFT) er en algoritme som utfører følgende matematiske operasjon på funksjonen $f_k = f(t_k)$ og finner DFT koeffisientene (F_n):

$$F_n = \sum_{k=0}^{N-1} f_k \cdot \exp(-2\pi \cdot i \cdot n \cdot k/N) = \sum_{k=0}^{N-1} f_k \cdot (i \cdot \sin(2\pi \cdot n \cdot k/N) + \cos(2\pi \cdot n \cdot k/N))$$

$$= i \sum_{k=1}^N f(t_n) \cdot \sin\left(\frac{2\pi \cdot n \cdot k}{N}\right) + \sum_{k=1}^N f(t_n) \cdot \cos\left(\frac{2\pi \cdot n \cdot k}{N}\right). \quad (40)$$



Figur 8: Figuren illustrerer operasjonene som gjøres ved en diskret Fourieroppløsning av et signal.

Operasjonen er illustrert i Fig. 8. Den digitaliserte verdien av funksjonen er i hvert punkt multiplisert med en cosinus (eller sinus) funksjon med frekvens $\omega_n = 2\pi \cdot \frac{n}{T} = \omega_0 \cdot n$ og deretter blir alle produktene er summert. Rekken av tall (vektoren) kalles den diskrete Fouriertransformerte av funksjonen.

Vi vil vise sammenhengen mellom DFT og FT.

Et signal logges i en bestemt tid, kalt T . Antallet digitaliseringer er N , slik at tiden mellom hver digitalisering er $T_s = T/N$. Tidspunktene for samplingene blir $t_k = kT_s$, der $k = 0, 1, 2, \dots, (N-1)$. Vi definerer da en variabel frekvens:

$$\omega_n = 2\pi \cdot \frac{n}{T} = \omega_0 \cdot n, \quad (41)$$

der $\omega_0 = \frac{2\pi}{T}$ kalles grunnfrekvensen og n er et heltall. Vi vil se hva som skjer når vi approksimerer FT med følgende uttrykk, der $n = 0, 1, \dots, (N-1)$:

$$\begin{aligned} F(\omega_n) &= \int_0^T \exp(-i \cdot \omega_n t) \cdot f(t) dt \\ &\approx T_s \cdot \sum_{k=0}^{N-1} \exp(-2\pi \cdot i \cdot n \cdot k/N) \cdot f(t_k) = \frac{T}{N} \cdot F_n, \end{aligned} \quad (42)$$

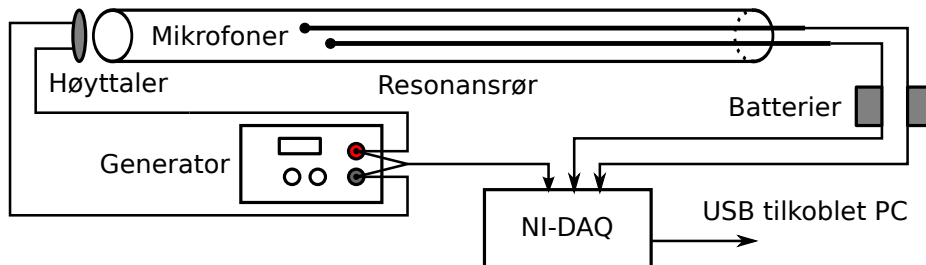
siden F_n nettopp var definert som denne summen. Altså er Fourierkoeffisientene de samme som DFT koeffisientene, bortsett fra en tallfaktor.

2 Praktisk utførelse

Punkt 1. Måling av lyd hastighet ved bruk av pulser (klikkelyd).

Utstyret kan settes opp som vist i Figur 9 og du kan gå fram som følger:

- Sett inn to trykkmålere (mikrofoner) inn i røret i forskjellig avstand fra høyttaleren. Disse har egne batterier som må skrues på. Utgangssignalene fra mikrofonkretsene sendes inn på inngang 1 og 2 på data innsamlingskortet (ett National Instrument data aquisition card; kalt NI-DAQ).



Figur 9: Apparatuoppsettet.

- Mål spenningen mellom utgangspolene med lav motstand på funksjonsgeneratoren med et multimeter, denne skal alltid være under 0.5 V (for å finne hvor mye under 0.5 V; bruk ørene og still lyden til noe som er behagelig å høre på)¹.
- Koble utgangspolene (de med lav motstand) på funksjonsgeneratoren til høyttaleren. Sett frekvensen til ca 1 Hz og bruk firkantsignal. Pass på at utgangsspenningen ikke er mer enn 0.5 V og kontroller dette med multimeteret. Når spenningen til høyttalermembranen plutselig endres, vil den trekkes ett stykke til siden og utføre en dempet svingning før den faller til ro rundt den nye likevektsposisjonen. Denne svingningen vil forplante seg med lydhastighet inne i røret.
- Koble utgangssignalet fra funksjonsgeneratoren til kanal 3 på NI-DAQ kortet. Dette signalet brukes som et triggersignal (startsignal) for datainnsamlingen. Idet signalet stiger fra null, eller når høyttalermembranen plutselig trekkes tilside, startes datainnsamlingen.
- Åpne Matlab og bruk programmet *acquire.m* (se Appendiks I) som samler data fra mikrofonene. For å se hvordan dette programmet skal brukes kan man skrive *help acquire* i Matlab, (husk å lage en mappe for å lagre dataene i). Loggfrekvensen settes til 50000 Hz, og signalene måles i 30 millisekunder. Du kan godt endre denne verdien og se hva som skjer; for eksempel at varighet for datainnsamling settes til 50 millisekunder.

Bruk tre forskjellige mikrofonavstander og regn ut lydhastigheten, som er

$$c = \frac{\Delta x}{\Delta t} = \frac{x_2 - x_1}{t_2 - t_1}, \quad (43)$$

der x er mikrofonposisjoner og t er tidspunktene der endringen i lydtrykket (klikket) treffer mikrofonen.

Den transiente lydbølgen (klikket) vil reflekteres fra enden av røret og vil også bli målt av mikrofonene. Du kan også benytte refleksene til å finne lydhastigheten.

Punkt 2. Lydhastighet og stående bølger.

Når den ene åpningen av røret eksiteres med en harmonisk bølge, vil lydfeltet inne i røret bestå av en stående og løpende bølge; se likn. (31).

$$\xi(x, t) = 2\xi_0^r \cdot \sin(kx) \cdot \cos(\omega t) + (\xi_0^i - \xi_0^r) \cdot \sin(kx - \omega t). \quad (44)$$

Dette uttrykket kan omformes til å bli følgende:

$$\xi(x, t) = A \cdot \sin(\omega t - \varphi), \quad (45)$$

der

$$A = \sqrt{(\xi_0^i + \xi_0^r)^2 \cdot \sin^2(kx) + (\xi_0^i - \xi_0^r)^2 \cdot \cos^2(kx)} \quad (46)$$

og

$$\tan \varphi = \frac{(\xi_0^i - \xi_0^r) \cdot \cos(kx)}{(\xi_0^i + \xi_0^r) \cdot \sin(kx)}. \quad (47)$$

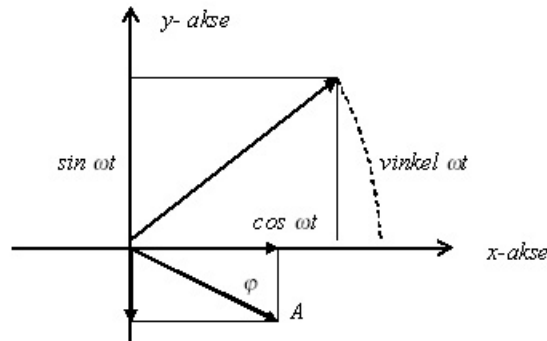
¹Pass på at amplituden er skrudd ned på null før du slår den på. Høyttaleren er liten og tåler bare effekten $P = 0,1$ W. Motstanden R i spolen som trekker høyttalermembranen att og fram er 8Ω , og det medfører at spenningen inn på høyttaleren fra generatoren ikke må være større enn $U = \sqrt{R \cdot P} = \sqrt{8 \cdot 0,10} \text{ V} = 0,90 \text{ V}$ (Husk: $P = U \cdot I = U^2/R$).

Dette uttrykket kan tilpasses til målte lydtrykk; og A, ω og φ blir funnet av tilpasningsrutinen.

Omformingen kan gjøres på følgende vis: Det første uttrykket omskrives til:

$$\begin{aligned}\xi(x, t) &= 2\xi_0^r \cdot \sin(kx) \cdot \cos(\omega t) + (\xi_0^i - \xi_0^r) \cdot [\sin(kx) \cdot \cos(\omega t) - \cos(kx) \cdot \sin(\omega t)] \\ &= (\xi_0^i + \xi_0^r) \cdot \sin(kx) \cdot \cos(\omega t) - (\xi_0^i - \xi_0^r) \cdot \cos(kx) \cdot \sin(\omega t)\end{aligned}\quad (48)$$

Vi tenker oss at vinkelen ωt har enhetslengde og roterer i xy -planet; se Fig. 10. Her er $\cos \omega t$ og $\sin \omega t$ lengdene av denne enhetsvektoren langs x - og y -aksene, henholdsvis. Videre blir disse lengdene forkortet/-forlenget med faktorene $(\xi_0^i + \xi_0^r) \cdot \sin(kx)$ og $-(\xi_0^i - \xi_0^r) \cdot \cos(kx)$, henholdsvis (vist med tykke piler langs x og negativ y -akse). Lydtrykket settes seg sammen av disse to komponentene til vektoren A , og dermed blir størrelse og fasevinkel gitt av uttrykkene over. $\xi(x, t) = A \cdot \cos(\omega t - \varphi) = A \cdot \sin(\omega t - \varphi + \frac{\pi}{2})$ er komponenten langs x -aksen.



Figur 10: Lydtrykket vist i et roterende diagram.

Dette er et forslag til framgangsmåte:

- Eksiter røret med et sinussignal med frekvens på 500 Hz.
- Sett først mikrofonene i samme posisjon inne i røret, for eksempel 15 cm fra høyttaleren og mål lydtrykkene ved bruk av `acquire.m` i Matlab (lag en ny mappe til disse dataene og sett triggering til **false**, hvorfor?). Programmet kommer til å spørre om posisjonene til mikrofonene. Disse posisjonene blir lagret i datafilene. Gjenta dette når du systematisk endrer posisjonen til den ene mikrofonen med 1 cm til den kommer til for eksempel 60 cm. Hvis du gjør feil er det bare å stoppe programmet og slette datafilen som var feil. Start programmet igjen og ta opp dataene på nytt.
- Nå kan amplituder, faser og frekvenser finnes ved sinustilpasninger. Programmet som gjør dette heter `tilpass.m` (se Appendiks II). For å få hjelp til programmet, skriv `help tilpass` i Matlab. Programmet lagrer alle dataene i en enkelt `.mat` fil.
- Nå som amplituder, faser og frekvenser er funnet, er det mulig å finne amplituden til den stående bølgen som funksjon av posisjonen i røret. For å gjøre dette trenger vi å finne tre parametre; bølgetallet, og amplitudene på innfallende og reflektert bølge. Til dette kan `m`-filen `behandling.m` brukes. Denne filen består av pseudokode, som må skrives om til vanlig Matlabkode. Tanken er å arbeide etter følgende prinsipp: Kvadratavviket, som er summen av kvadratene på differenser mellom målte og teoretisk verdier, blir minimalisert (ved hjelp av `fminsearch`) ved å variere tilpasningsparametrene (ξ_0^i, ξ_0^r og k). Disse settes ved å gjøre kvalifisert gjetning. Dessuten må du bruke en fjerde tilpasningsparameter, x_0 , som er nullpunktet (origo) for bølgetallet. Grunnen til at vi trenger x_0 er at grensebetingelsen ved høyttaleren ikke har noen enkel form. Praktisk betyr dette at x i uttrykket som skal optimeres på må byttes ut med $x - x_0$. En god startbetingelse for x_0 kan være -0.05 m. Merk at ξ_0^i og ξ_0^r bare forekommer i lag inne i kvadratiske uttrykk, og derfor kan `fminsearch` noen ganger bytte om på de to.
- Kontroller overensstemmelsen ved å framstille målte og teoretiske verdier i samme figur. Finn bølglengden og lydshastigheten ved bruk av relasjonene:

$$\lambda = \frac{2\pi}{k} \quad \text{og} \quad c = \frac{\omega}{k} = f \cdot \lambda$$

Punkt 3: Fourieromvending og bruk av støy.

Til dette eksperimentet brukes et resonansrør som er åpent i begge ender og hvit støy. Hvit støy er kjennetegnet ved at den inneholder alle frekvenser. Hvis denne lyden sendes inn i et resonansrør vil bare frekvensene i støyen som samsvarer med resonansfrekvensen i røret bli forsterket. Det er dette vi skal se ved å gå fram som følger:

- Plasser mikrofonene inne i røret.
- Bruk signalgeneratoren og utsett røret for et harmonisk signal.
- Start så *acquire.m*, ved å bruke **1 kanal** og ingen triggering, i tillegg til å logge i **minst 1 s**.
- Etter at Matlab har lest og regnet ut FFT skal figurene vises. En av figuren viser FFT av signalet fra kanal 1.
- Studer dette spekteret.

Nå skal røret utsettes for hvit støy:

- Støysignalet lages ved hjelp av trykkluft.
- Åpne Matlab og bruk *acquire.m* (samme innstillinger som for det harmoniske signalet) til å logge fra mikrofonen når røret eksiteres med trykkluft.
- Noter resonansfrekvensene til røret og regn ut lyd hastigheten, se ligning 33.

Punkt 4: Bestem resonansfrekvensene til en orgelpipe som eksiteres med støy.

En orgelpipe har to åpene ender, og har veldefinerte resonansfrekvenser. Disse skal finnes ved hjelp av Fourieromvending.

Fremgangsmåten er som følger:

- Bruk samme m-fil (*acquire.m*) og innstillinger som i forrige punkt.
- Ta mikrofonen ut av røret og legg den på bordet.
- Ta opp lyden fra en orgelpipe med Matlab. Orgelpipen blåses på med trykkluft.
- Finn resonansfrekvensene og finn hvilken note det tilsvarer, se tabell 1
- Hvorfor stemmer ikke ligning 33 helt her?

Tabell 1: Frekvenser for de forskjellige notene

Note	Omtrentlig frekvens (Hz)
C	262
C#	277
D	294
D#	311
E	330
F	349
F#	370
G	392
G#	415
A	440
A#	466
B	494

Appendiks I - acquire.m

```
1 function acquire(lydlab,numChan,trigging,sampleFreq, duration,saveFolder)
2
3 % ACQUIRE er en funksjon for å ta opp data fra en DAQ. Programmet tar fire
4 % argumenter lab, numChan, trigging, sampleFreq og saveFolder.
5 % - lydlab: en string som bestemmer om det er lydbølgelab (true) eller
6 %       vannbølgelab (false)
7 % - numChan: antall kanaler som det skal måles på (1 eller 2)
8 % - trigging: om datainnsamlingen skal trigges (true eller false)
9 %       NB: true forutsetter at det er 2 kanaler som det tas opp på
10 % - sampleFreq: hvor ofte det skal måles i Hz (50000 eller 200)
11 % - duration: hvor lenge det skal måles i sekunder (0.02 eller 5)
12 % - saveFolder: hvor dataene skal lagres. Denne bør være tom, og
13 %       forskjellig mellom hver gang funksjonen kjøres
14 %
15 % Eksempel på bruk (for lydbølgelab)
16 %   acquire(true,2,true,50000,0.02,'D:\Documents and Settings\localuser\Desktop\Data1')
17 %
18 % Skrevet av Pål Ellingsen, v 07.09.2012
19 % delvis basert på oppsett av Iver Bakken Sperstad
20
21 fortsett = true;
22
23 while fortsett % for å ta flere målinger på rad
24
25     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
26     try % Bruker try catch for å finne rett enhet
27         AI=analoginput('nidaq','Dev3');
28     catch exception
29         try
30             AI=analoginput('nidaq','Dev2');
31         catch exception2
32             AI=analoginput('nidaq','Dev1');
33         end
34     end
35     %identifiserer "device"(Et National Instrument datainnsamlingkort (DAQ))
36
37     set(AI,'InputType','Differential');
38     %spesifiserer at det skal måles differensielt(spenningen mellom to ledninger)
39
40     Chan=addchannel(AI,1:(numChan+trigging));
41     % Antall kanaler som logges pluss en eventuell trigger kanal
42
43     AI.Channel.InputRange=[-5 5];
44     %spesifiserer måleområdet; mellom -1.5 og 1.5 V
45
46     set(AI,'SampleRate',sampleFreq);
47     %setter samplingsrate
48
49     sampleFreq=get(AI,'Samplerate');
50     % henter ut hva det virkelige samplingsraten er
51     set(AI,'SamplesPerTrigger',duration*sampleFreq);
52     %maskiner avsetter plass til datastreng
53
54     if trigging % setter opp trigging hvis det er ønsket
55
56         set(AI,'TriggerChannel',Chan(3))
57         set(AI,'TriggerType','software')
58         set(AI,'TriggerCondition','Rising')
59         set(AI,'TriggerConditionValue',0.00)
60         start(AI); % starter opptak
61
62     else % manuell trigging
63         set(AI,'TriggerType','Manual');
64         start(AI); % starter opptak
```

```

65     trigger(AI) % trigger opptak
66 end
67
68 data=getdata(AI);
69 %dette er datastrengen
70
71 delete(AI); % sletter
72 clear AI;
73
74 %%%%%%%%% Databehandling %%%%%%%%%%%%%%%
75
76 s1=data(:,1:numChan); % lager en vektor for hver del av dataen
77
78 n=1:length(data);
79 tid=n./sampleFreq;
80
81 %samplingstidspunktene bergnes
82
83 figure(1) % plotter den målte dataene
84 clf(1)
85 plot(tid,s1(:,1),'b')
86 if numChan >1
87     hold on;
88     plot(tid,s1(:,2),'r')
89     hold off;
90     legend('Channel 1','Channel 2');
91 xlabel('Tid (s)')
92 ylabel('Spenning (V)')
93 end
94
95 filename = ['\ ',datestr(now,'HH_MM_SS'), '_samp_',num2str(sampleFreq),...
96     '_dur_',strrep(num2str(duration),'.','_')];
97 % lager et filnavn (blir noe som 15_45_samp_50000_dur_0.02
98
99
100 if numChan ==1
101     y1=fft(s1(:,1));
102
103     % det gjøres en Fourieromvendning av signalet
104     n1=length(s1(:,1));
105
106     %finner lengden av datastrengen
107     abs1=y1(1:n1/2).*conj(y1(1:n1/2));
108
109     %regner ut absoluttverdien av den Fourieromvendte (halve lengden)
110     f1=(0:(n1/2-1))*(sampleFreq/n1);
111
112     %beregner tilhørende frekvenser
113
114     figure(2)
115     clf(2)
116     plot(f1,abs1)
117     xlabel('f (Hz)')
118     title('FFT')
119     %frekvensspekteret plottes
120 end
121
122 if lydlab && not(triggering) && numChan~=1 % for å lese inn posisjon av mikrofon
123     mic1 = input('Posisjon av mikrofon 1: ');
124     mic2 = input('Posisjon av mikrofon 2: ');
125     filename = [filename, '_mic1_',num2str(mic1),'_mic2_',num2str(mic2)];
126 % legger til x1 og x2 til filnavn
127 elseif not(lydlab) % For å lese inn spenning
128     V = input('Spenning: ');
129     filename = [filename, '_spenning_',num2str(V)];
130 end
131
132 clear Chan exception

```

```
133
134     save([saveFolder,filename, '.mat'])
135     % lagrer alle variablene i en mat fil
136
137     igjen = input('Ny måling med samme parametre? [y]','s');
138     if isempty(gjgen) || isequal(gjgen,'y')
139     else
140         fortsett = false;
141     end
142 end
143
144 end
```

Appendiks II - tilpass.m

```
1 function tilpass(dataFolder,saveFile)
2 % Utfører minste-kvadrat-tilpassing til ein sinusfunksjon på to lydsignal
3 % for å finne faseforskjellen mellom dei (modulo  $2\pi$ ).
4 % Tar to input: dataFolder og saveFolder.
5 % dataFolder: en string med mappen som inneholder dataene.
6 % saveFile: en string med filen som inneholder dataene.
7 %
8 % Eksempel på bruk: tilpass('Data1','Data1\Res\res.mat')
9 %
10 % Skrevet av Pål Ellingsen, v 16.09.2011 delvis basert på
11 % sinusregresjon, skrevet av Iver Bakken Sperstad, som igjen var basert på
12 % tilpassingsrutine for Cavendish-lab i Mekanisk fysikk av Frantz Stabo-Eeg
13 % og Eirik Glimsdal (2006–2008).
14
15 files = dir([dataFolder,'\*.mat']) %finder alle filene i mappen med ending .mat
16
17 for k=1:length(files) % går gjennom alle filene og tilpasser på hver av de
18
19     % Les data fra fil (antek ei kolonne for kvart signal)
20
21     files(k).name
22     filename = [dataFolder,'\ ',files(k).name]
23     if isempty(who('-file', filename ,'-var','lydlab')) % tester at det er rett fil
24         % ikke rett
25         if input(['Filen ', filename,...
26                 ' inneholder ikke måledat. Vil du fortsette til neste fil? y/n'])=='y';
27             continue
28         else
29             return
30         end
31     end
32
33     load(filename); % laster inn alle variablene
34
35     % Initialiserer tilpassingsparametrar
36     % (antek signal på forma  $s(t) = S + A\sin(2\pi f t + \phi)$ )
37
38     S = [0 0]; % middelerdi for signalet
39     A = [0 0]; % amplitude
40     f = [0 0]; % frekvens
41     phi = [0 0]; % fase
42
43     % Går gjennom eitt av signala om gongen
44     for signal = 1:numChan
45
46         s = s1(:,signal); % plukkar ut rett signal
47         len = length(s);
48         t = (1:len)';
49         % tidsvektor med heiltal; tek ikkje omsyn til samplingsfrekvens før på slutten
50
51         % Initiell tilpassingsrutine, steg 1
52         minS = min(s);
53         maxS = max(s);
54         meanS = 0.5*(minS + maxS);
55         % finn omtrentleg middelerdi for signalet (dvs. gitt at talet på periodar er heiltal)
56
57         prevAboveMean = ( s(1) > meanS );
58         % finn ut om signalet startar over (= 1) eller under (= 0) middelerdien
59         crossMean = [];
60         % kryssingspunkt, dvs. når signalet kryssar middelerdien
61
62         % Initiell tilpassingsrutine, steg 2: Gå gjennom signal og lokaliser kryssingspunktta
63         for i=2:len
64             aboveMean = ( s(i) > meanS ); % er signalet over eller under middelerdi nå?
```



```

65         if ( aboveMean ~= prevAboveMean )
66             % dersom signal går frå å vere over(under) til å vere under(over) middelvei, ...
67                 crossMean = [crossMean i];           % ...lagre som kryssingspunkt
68                 prevAboveMean = aboveMean;
69             end
70         end
71
72         % Initiell tilpassingsrutine, steg 3:
73         % Finn ut kor mange periodar signalet består av (med korrigering av støy i signalet)
74
75         halfPeriod = crossMean(2:end) - crossMean(1:end-1);
76         % lengda på halvperiodar, dvs. avstand mellom kryssingspunkt
77
78         numHalfPeriods = length(halfPeriod);
79
80         badHalfPeriods = find( halfPeriod < max(halfPeriod)/2 );
81         % dersom ei halvperiode er uvanleg kort skuldast det
82         % truleg berre støy rundt kryssingspunkt, ...
83
84         numHalfPeriods = numHalfPeriods - length(badHalfPeriods);
85         % ...så ikkje ta omsyn til desse i berekning av periodetal
86
87         meanPeriod = 2*sum(halfPeriod)/numHalfPeriods;           % estimerer lengda på ein periode
88
89         if (numHalfPeriods < 3)
90             disp('Feil: Klarer ikkje å finne ein heil periode i signalet. ');
91             return;
92         end
93
94         % Set startverdiar for tilpassingsrutine basert på initiell tilpassingsrutine
95         S0 = meanS;
96         A0 = maxS - meanS;
97         f0 = 1/meanPeriod;
98
99         phi0 = -2*pi*f0*crossMean(1);
100        % baserer fase-estimat på fyrste registrerte kryssingspunkt
101        if ( s( crossMean(1) + round(0.25*meanPeriod) ) < S0 )
102            % dersom signalet er på veg nedover (og ikkje oppover) ved fyrste kryssingspunkt...
103            phi0 = phi0 - pi;
104            % ...så må fassen korrigerast tilsvarande ein halv periode
105        end
106
107        x0=[S0,A0,2*pi*f0,phi0];           % startpunkt for tilpassingsiterasjonar
108
109        modelFunc = @(x,t) x(1) + x(2).*sin(x(3).*t +x(4)); % lager modellfunksjon
110
111        sumSquareDeviation=@(x,s,t) sum((s-modelFunc(x,t)).^2 ); % regner ut kvadratavviket
112        % Kurvetilpassing ved optimalisering av tilpassingsparametrar
113        [xFit fval] = fminsearch(@(x) sumSquareDeviation(x,s,t), x0);
114
115        % Lagrar tilpassingsparametrar
116        S(signal) = xFit(1);
117        A(signal) = xFit(2);
118        f(signal) = xFit(3)/(2*pi)*sampleFreq; % tek omsyn til samplefrekvens her
119        phi(signal) = xFit(4);
120        % NB: tilpassingsparameteren er ikkje nødvendigvis avgrensa til eitt 2\pi-intervall
121
122        % Bereknar kurve basert på tilpassingsparametrar
123        sFit=modelFunc(xFit,t);
124
125        % Plottar signal (tidsakse skalert med samplingsfrekvens,
126        % skalerer signal slik at middelvei blir null for både)
127        figure(1)
128        if (signal == 1)
129            plot(t/sampleFreq,s-S(1), 'b. ');
130            hold on
131            plot(t/sampleFreq,sFit-S(1), 'r—')
132        elseif (signal == 2)

```

```

133         plot(t/sampleFreq,s-S(2),'Color',[0 0.5 0],'Marker','.', 'LineStyle','None');
134         plot(t/sampleFreq,sFit-S(2),'m—')
135         hold off
136         xlabel('t [s]')
137         ylabel('Luftrykk (skalert og med vilkårlige einingar)')
138     end
139
140 end
141
142 legend('Signal 1','Kurvetilpassing til signal 1','Signal 2','Kurvetilpassing til signal 2')
143
144 phi=mod(phi,2*pi); % Forandrer phi slik at den blir mellom [0,2\pi)
145
146 % Skriv resultat til skjerm
147 disp(' ')
148 disp('*** Tilpassingsparametrar for hhv. signal 1 og 2 ***')
149 disp(' ')
150 disp(['Amplitude: ' num2str(A)])
151 disp(['Frekvens:   ' num2str(f)])
152 disp(['Fase:      ' num2str(phi)])
153
154 figure(1)
155
156 serBraUt = input('Er tilpassnigen bra? y/n [y]','s');
157 if isempty(serBraUt) || serBraUt == 'y'
158 else
159     return
160 end
161
162 if isempty(whos('mic1')) % tester om mic1 finnes
163     resultater(k,:) = [A,f,phi];
164     formatResultater = 'mic1, mic2, A1 ... An, f1... fn, phi... phin';
165 % viser hvordan dataene er lagret
166 else
167     resultater(k,:) = [mic1,mic2, A,f,phi];
168     formatResultater = 'mic1, mic2, A1 ... An, f1... fn, phi... phin';
169 % viser hvordan dataene er lagret
170 end
171
172 end
173
174 save(saveFile,'resultater','formatResultater') % lagerer dataene.
175
176 end

```

Appendiks III - behandling.m

```
1 % Denne filen inneholder det som skal til for å kunne analysere dataene
2 % dere har tatt opp.
3
4 clear all % fjerner gamle variabler
5
6 %% Først må dataene lastes, bruk load til dette.
7 load(
8
9 %% Så kan dere plotte datene. Bruk plot for å gjøre dette.
10
11 X = resultater(:,kolonne);
12 Y = resultater(:,kolonne);
13 plot(X,Y)
14
15 %% Så skal dere tilpasse dataene. Til det bruker vi fminsearch, som søker
16 %% etter minimum til funksjonen
17 % For å gjøre dette trenger dere fire startbetingelse:
18 epsI0 = % Innkommende amplitude, som kan estimeres fra plottet av den stående
19         % bølgen som dere har plottet over
20 epsR0 = % Reflektert amplitude, som kan estimeres fra den første oppgaven, som
21         % en prosent av epsI0
22 k0     = % Kan måles fra plottet av den stående bølgen
23 x0     =
24
25 c0 = [epsI0,epsR0,k0,x0]; % lager en vektor med alle startbetingelsesn
26
27 % så må modellen bestemmes, dette gjøres ved å lage en funksjon. Hvis
28 % modellen var en rett linje ville det gjøres slik:
29 % modell = @(c,x) c(1)+c(2).*x % her forteller @( ) hva som er variabler
30 % og etter den følger modellen. c er en vektor med alle konstantene som
31 % skal tilpasses. Det vil si at c er en vector med like mange elementer
32 % som c0, som betyr at c(1) = epsI, c(2)= epsR osv.
33 % Husk å bruke punktum foran *, / og ^.
34
35 modell = @(c,x)
36
37 % For å kunne bruke tilpassningen må man gi inn en funksjon som regner ut
38 % kvadratavviket. For den rette linjen vil dette være:
39 % avvik = @(c,s,x) sum((s-modell(c,x)).^2);
40 % hvor s er måledataene. Siden vi har en modell og måledataer, som vi kaller
41 % s, kan vi bruke den samme funksjonen som for den rette linjen her.
42
43 avvik = @(c,s,x) sum((s-modell(c,x)).^2);
44
45 % Definere måledataene og x koordinatet
46 s = resultater(:,
47 x = resultater(:,
48
49 %
50
51 options = optimset('MaxFunEvals',1e8); % Forandrer maks antall ganger
52 % funksjonen kan evalueres
53
54 % så kjøres tilpassningen
55 [cfit fval] = fminsearch(@(c) avvik(c,s,x) , c0) % tar inn avviksfunksjonen og startpunktene (c0)
56 % cfit er vektoren med resultatet, på samme form som c0
57 % fval er resultatet som avvik returnerer når optimeringen er ferdig
58
59
60 %% Plot deretter resultatene og regn ut lydshastigheten
61 % Husk at du kan nå evaluere løsningen ved å kalle modell med cfit og x som
62 % argumenter
```