FYS2130 regneoppgaver uke 08

Lyd

OBLIG innlevering med frist 17.03.2021, kl. 0900

Monday 22$^{\text{nd}}$ March, 2021, 12:30

OPPGAVE 1: Desibel

Vis matematisk at det er riktig å si: Å legge til X dB i lyden svarer til å multiplisere intensiteten til den opprinnelige lydbølgen med en bestemt faktor.

OPPGAVE 2: Gitarstreng

For denne oppgaven kan du bruke figur 7.24 for å en forklaring av de relevante delene på en gitar.

a) Lengden på den fri delen av strengene på en gitar (d.v.s. den delen som kan svinge) er 65 cm. Klemmer vi ned G-strengen i femte båndet, får vi en C. G-en har en frekvens om lag $196,1$ Hz og C-en om lag $261,7$ Hz. Hvor må det femte båndet være plassert på gitarhalsen?

b) Bruk svar fra forrige oppgave. For hver halvtone vi går opp fra der vi er, må frekvensen øke med en faktor $1,0595$. Beregn posisjonen til første båndet, og til sjette båndet. Er avstanden mellom båndene (målt i antall millimetre) identiske langs gitarhalsen? Vis at avstanden mellom båndene er gitt ved 0,0561 ganger lengden til strengen da den var klemt inn i forrige bånd.

OPPGAVE 3: Doppler

Anta at du kjører bil i 60 km/t og hører at en politibil med sirener nærmer seg bakfra og kjører forbi. Du merker den vanlige endringen i lyd idet bilen passerer. Anta at politibilen kjører i 110 km/t og at den øvre frekvensen i sirenen har en frekvens på 600 Hz dersom vi hadde lyttet til sirenen i politibilen. Hvilke frekvenser opplever vi å høre før og etter at politibilen har kjørt forbi oss?

OPPGAVE 4: Grunntone versus harmoniske (frivillig)

Denne oppgaven er frivillig, for spesielt interesserte. Det vil si at den *ikke regnes med* når vi vurderer godkjent/ikke-godkjent.

Vi ønsker å sjekke påstanden i boka om at vi kan "ta bort grunntonen fra det harmoniske spekteret fra et opptak av et instrument, og likevel få en lyd som høres nesten lik ut". Du kan teste dette som følger:

- Ta utgangspunkt i koden fra hjemmelabben (uke 8 - del 3) som gjennomgått i forelesning. Bruk f.eks. opptakket "gitarelo.wav", eller egne opptak.
- Skriv kode som setter alle Fourier-koeffisienter X_k som tilsvareren frekvenser lavere enn en terskel, f_{min}, til 0.
- Inverstransformer X_k tilbake til en tidsserie, og spill av denne, for forskjellige, økende verdier av f_{min}, som tar bort én og én harmonisk komponent.
- Kommenter hvordan lyden høres ut i forhold til den opprinnlige lyden.

Noen tips for implementasjonen:

- Husk foldingsssymmetri når du nuller ut frekvenser.
- Du kan spille av på flere måter. En måte er å skrive tidsserien tilbake til en .wav fil igjen med "write" kommandoen.
- Dobbelsjekk ved å først regenerere tidsserien med frekvensinnhold inntakt, og sammenlign denne lyden med den opprinnlige, for å sjekke at du gjør inverstransformen korrekt (f.eks. normaliseringer).
- NB: ikke ha volum på fullt når du eksperimenterer med å spille av egengenererte tidsserier!

FASIT FOR OPPGAVE 1

Desibel-skalaen er logaritmisk, så en endring i intesitet gitt i desibel vil alltid tilsvare en gitt faktors endring av det opprinnelige signalet. Dette kan vises slik:

Vi har en frekvens I ganget med en konstant faktor k.

$$\begin{aligned} L(kI) &= 10\log(k\frac{I}{I_0}) = 10[\log(\frac{I}{I_0}) + \log(k)] \\ &= 10\log\frac{I}{I_0} + 10\log k = L(I) + XdB \end{aligned}$$

FASIT FOR OPPGAVE 2

a) Hastigheten til bølgen på en streng kan defineres som (se for eksempel fra ligning (7.5) i boka)

$$v = 2Lf$$

der L er lengden på strengen og f er frekvensen på en bølge på strengen. v er konstant for alle bølger på en bestemt gitarstreng, ettersom hastigheten bare er gitt fra massetettheten og spenningen i snoren. Størrelsen Lf er altså bevart for en gitt streng. Vi har da at

$$L_G f_G = L_C f_C \quad \Rightarrow \quad L_C = L_G \frac{f_G}{f_C} = 0,65 \text{ m} \cdot \frac{196,1 \text{ Hz}}{261,7 \text{ Hz}} = 0,487 \text{ m}$$

Det femte båndet er plassert $0,487$ m oppe på gitarhalsen.

b) Vi definerer f_0 som den laveste frekvensen på en streng, når ingen av båndene holdes inn. Vi kan da skrive de øvrige frekvensene som

$$f_n = (2^{1/12})^n = (1,0595)^n f_0$$

Ved relasjonen fra forrige oppgave kan vi sette inn $f_n = f_0 \frac{L_0}{L_n}$ og regne ut forholdet mellom lengdene på båndene.

$$f_0 \frac{L_0}{L_n} = (1,0595)^n f_0$$
$$L_n = \frac{L_0}{(1,0595)^n} = (0,9439)^n L_0$$

Avstanden på båndene er ikke konstant, ettersom dette ikke er en lineær funksjon.

Vi kan eventuel skrive avstanden som

$$L_n = 0,9439 \cdot L_{n-1}$$

Avstanden mellom to bånd blir da

$$L_n - L_{n-1} = 0,9439 \cdot L_{n-1} - L_{n-1} = (1 - 0,9439) \cdot L_{n-1} = 0,0561 \cdot L_{n-1}$$

FASIT FOR OPPGAVE 3

Hvis et objekt med hastighet v_k sender en lyd med frekvens f_k, som måles av en observatør med hastighet v_0, vil observatøren måle en frekvens på

$$f_0 = \frac{v + v_0}{v - v_k} f_k$$

der v er lydhastigheten i mediet de befinner seg i. Her har observatøren en hastighet $v_0 = -60$ km/t $= -16,67$ m/s, og kilden en hastighet $v_f = 110$ km/t $= 30,56$ m/s. Lydhastigheten i luft er ≈ 343 m/s. Vi får da en observert frekvens når bilen kjører imot på

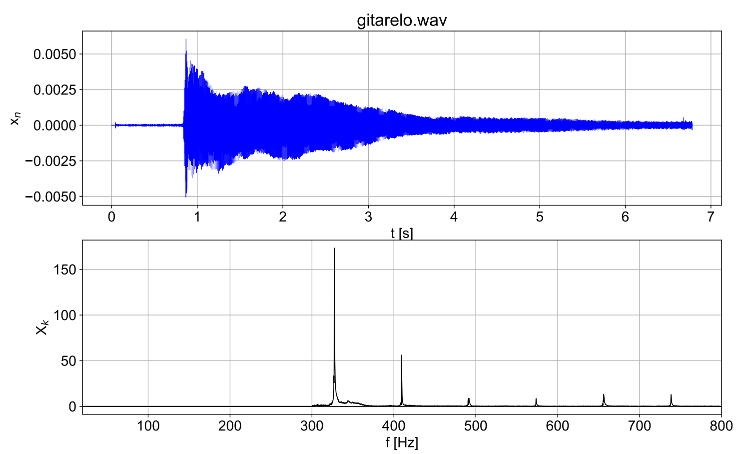
$$f_0 = \frac{343 - 16,67}{343 - 30,56} 600 \text{ Hz} \approx 626,7 \text{ Hz}$$

Når politibilen har passert, snur vi fortegn på begge hastigheten (Vi beveger oss nå mot observatøren, som beveger seg bort fra oss).

$$f_0 = \frac{343 + 16,67}{343 + 30,56} 600 \text{ Hz} \approx 577,7 \text{ Hz}$$

FASIT FOR OPPGAVE 4

En kode for å studere dette er vedlagt. I denne koden sammenlignes den originale lyden med en ny lyd der vi har tatt bort ikke bare én, men de tre første harmoniske. For meg høres den resulterende lyden ut som den samme tonen, bare litt tørrere. Frekvensspekteret ser slik ut:




```
%reset -f
import numpy as np
import matplotlib.pyplot as plt
from scipy.io import wavfile
from scipy import signal
import os
from scipy.io.wavfile import write

#filename = '/Users/eadli/Dropbox/COURSES/FYS2130_2021/
    FORELESNINGER/U8/samples/klangstavE.wav'
#filename = '/Users/eadli/Dropbox/COURSES/FYS2130_2021/
    FORELESNINGER/U8/samples/dameEhi.wav'
#filename = '/Users/eadli/Dropbox/COURSES/FYS2130_2021/
    FORELESNINGER/U8/samples/blokkfloyteE.wav'
#filename = '/Users/eadli/Dropbox/COURSES/FYS2130_2021/
    FORELESNINGER/U8/samples/pianoE.wav'
filename = '/Users/eadli/Dropbox/COURSES/FYS2130_2021/FORELESNINGER
    /U8/samples/gitarelo.wav'

nameonly = filename[67:]
```

```
# tidsserie
samplerate, data = wavfile.read(filename)
x_n = data[:, 0] # select one out of two channels
f_samp = samplerate
# select time interval, and down sample
N_temp = len(x_n)
T_temp = N_temp / f_samp # s
t1 = 0
t2 = T_temp
N_subsample = 8
#x_n = data[np.int_(t1*samplerate):np.int_(t2*samplerate):
    N_subsample, 0]
#f_samp = samplerate//N_subsample
# parametre
N = len(x_n)
T = N / f_samp # s
print(f"samplerate = {f_samp} Hz")
print(f"T = {T} s")
dt = 1/f_samp
t = dt*np.linspace(0., N, N)

# playing original sample
os.system("afplay " + filename + "&")

# calc DFT via FFT
X_k = (1/N)*np.fft.fft(x_n)
freq = np.fft.fftfreq(N, dt)

#
# HERE starts the kode to remove harmonics
#

# we remove this when we want to inverstransform, since we need to
    also zero the folder part of the spectrum
#X_k = X_k[:N//2]
#freq = freq[:N//2]

X_k_mod = X_k
f_min = 300 # remove everything below this frequency
# remove all content up to a certain frequency if the below is set
    to "1"
if 1:
    freq_0_idx = np.nonzero(freq < f_min)
    X_k_mod[freq_0_idx] = 0

x_n_mod = np.real(np.fft.ifft(X_k_mod)) # inverstransform back to
    time series

# write back to .wav in order to use the play sample function we
    know
mod_filename = '/tmp/x_k_mod.wav'
x_n_scaled = np.int16(x_n_mod/np.max(np.abs(x_n_mod)) * 32767) #
    normalise amplitude
write(mod_filename, 44100, x_n_scaled)

input("Press any key to listen to the modified sample.")
```

```python
os.system("afplay " + mod_filename + "&")

#
# HERE ends the kode to remove harmonics
#



idxmax = np.argmax(X_k)
print(f" peak = {freq[idxmax]} Hz")

# plot tidserie og FFT
font = {'family' : 'Arial','weight':'normal','size'   : 16}
plt.rc('font', **font)
#
fig, ax = plt.subplots(2,1, figsize = (13, 8))
ax[0].grid(1)
ax[1].grid(1)
ax[0].plot(t, x_n_mod, color='blue', linestyle='solid', linewidth
    =0.5)
ax[0].set_xlabel('t [s]')
ax[0].set_ylabel('x$_n$')
#
ax[1].plot(freq, np.abs(X_k[:N]), color='black', linewidth=1.0)
ax[1].set_xlabel('f [Hz]')
ax[1].set_ylabel('X$_k$')
ax[1].set_xlim([20, 800])
#ax[1].set_xlim([150, 5700])
ax[0].set_title(nameonly, )
plt.show()
#fig.savefig("/Users/eadli/Dropbox/COURSES/FYS2130_2021/
    FORELESNINGER/U8/samples/opptak_%s.pdf" %nameonly, format='pdf
    ')
```