ØV9 — DFT 2

Innleveringsfrist: **30. oktober** 2020.

Ukeoppgavene skal løses selvstendig og vurderes i øvingstimene. Det forventes at alle har satt seg inn i fagets øvingsopplegg og godkjenningskrav for øvinger. Dette er beskrevet påhjemmesiden til IN3190:
`http://www.uio.no/studier/emner/matnat/ifi/IN3190/h20/informasjon-om-ovingsopplegget/`

Oppgave 1 — Topic: Windowing 5 Points

The Tukey window – also denoted cosine-tapered window – can be defined as follows:

$$w_{\text{tuk}}(t) = \begin{cases} \frac{1}{2}\left[1 - \cos\left(\left(t + \frac{T_0}{2}\right)\frac{2\pi}{rT_0}\right)\right] & \text{for } -\frac{T_0}{2} \le t < \left(-\frac{T_0}{2} + \frac{T_0}{2}r\right) \\ 1 & \text{for } \left(-\frac{T_0}{2} + \frac{T_0}{2}r\right) \le t \le \left(\frac{T_0}{2} - \frac{T_0}{2}r\right) \\ \frac{1}{2}\left[1 - \cos\left(\left(t - \frac{T_0}{2}\right)\frac{2\pi}{rT_0}\right)\right] & \text{for } \left(\frac{T_0}{2} - \frac{T_0}{2}r\right) < t \le \frac{T_0}{2} \end{cases}$$

where r is a parameter varying between 0 and 1 and T_0 is the window length.

a) Use Pyton, Matlab, Julia or any other of your favourite frameworks to:

   - plot the Tukey window sampled with $N = 100$ samples, for $T_0 = 1$ and $r = 0.25, 0.5, 0.75, 1$. 2 p.
   - plot the corresponding frequency spectrum.

   It could be useful to verify the correctness of your own Tukey function implementation using output from the Python / Matlab / etc. builtin Tukey window function.

b) For each r setting: Assess the mainlobe width and sidelobe level of the Tukey window frequency spectrum.

   What trend do you notice? How does this relate to the window shape? 2 p.

   Compare the nature of your results with the plots provided in the Matlab documentation: `https://www.mathworks.com/help/signal/ref/tukeywin.html`

c) What other windows flavours does Tukey correspond to for the cases $r = 0$ and $r = 1$? 1 p.

Oppgave 2 — DFT / FFT Vekt:3

For desimering-i-tid FFT algoritmen kan vi beregne en N-punkts DFT til sekvensen $x[n]$ fra to $N/2$-punkts DFTer:

$$\begin{aligned} X[k] &= X_e[k] + W_N^k X_o[k], \quad k = 0, 1, \dots, \frac{N}{2} - 1 \\ X\left[k + \frac{N}{2}\right] &= X_e[k] - W_N^k X_o[k], \quad k = 0, 1, \dots, \frac{N}{2} - 1. \end{aligned} \tag{1}$$

hvor $X_e[k]$ og $X_o[k]$ er DFTene til par- og oddetallssamplene til $x[n]$.

Den beregningsmessige kompleksiteten, dvs. orden av antall operasjoner, for direkt beregning av en N-punkts DFT ved å bruke definisjonen

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \quad k = 0, 1, \dots, N-1,$$

hvor de komplekse eksponensialfaktorene $W_N^{kn} = e^{-j2\pi kn/N}$ er lagret i en tabell, krever $O\left(N^2\right)$ komplekse operasjoner.

Fast Fourier Transform (FFT) algoritmer reduserer den beregningsmessige kompleksiteten til $O\left(N \log_2 N\right)$ operasjoner.

a) 4-punkts DFTer

La

$$x[n] = \{\underset{\uparrow}{1}, \tfrac{\sqrt{2}}{2}, 0, -\tfrac{\sqrt{2}}{2}, -1, -\tfrac{\sqrt{2}}{2}, 0, \tfrac{\sqrt{2}}{2}\}.$$

være en 8-punkts sekvens. Hva er de to 4-punkts sekvensene $x_e[n]$ (partall) and $x_o[n]$ (oddetall) som brukes i det første steget av FFT ved desimering-i-tid.

b) 2-point DFTs

Bruk Ligning (1) for å vise at de fire 2-punkts DFTene $X_{ee}[k]$, $X_{oe}[k]$, $X_{eo}[k]$ og $X_{oo}[k]$ beregnet i andre steg av FFT ved desimering-i-tid er gitt ved:

$$\begin{aligned} X_{ee}[k] &= \{0,\ 2\} \\ X_{oe}[k] &= \left\{0,\ \sqrt{2}\right\} \\ X_{eo}[k] &= \{0,\ 0\} \\ X_{oo}[k] &= \left\{0,\ -\sqrt{2}\right\}. \end{aligned}$$

c 8-punkts DFT

Beregn de to 4-punkts DFTene $X_e[k]$ og $X_o[k]$ ved å benytte deg av Ligning (1), og beregn så 8-punkts DFTen $X[k]$.

Hint: Samsvarer ditt endelige svar med det du ville forventet fra formen på/verdiene av inngangssekvensen?

Oppgave 3 — FFT Vekt:2

1. Peter ønsker å konvolvere to sekvenser $x(n)$, med lengde $N = 1024$, og h(n), med lengde L. Han lurer på hvilken metode han skal velge og spør Andreas om hjelp. For å være sikker på å få rett svar, spør Andreas dere:

   - For hvilken lengder L er det færre multiplikasjoner ved direkte beregning av konvolusjonen $x(n) * h(n)$ enn ved å utnytte radix-2 FFT algoritmen?

   I det siste tilfelle kan man finne konvolusjonen ved å ta invers DFT av produktet $X(k)H(k)$, der $X(k)$ er DFT av $x(n)$ og $H(k)$ er DFT av $h(n)$.

2. Henrykt over mange svar kommer Peter med enda flere spørsmål, nå direkte til dere. Han har en ny sekvens $y(n)$ som han er sikker på er kompleks og av lengde 1025 (dvs 1 mer enn $N = 2^v$). I stedet for å se bort fra siste datapunkt, vurderer han å null-utvide sekvensen slik at den får lengde $N = 2^w$ slik at radix-2 FFT-algoritmen kan brukes.

   - Hvor mange multiplikasjoner og addisjoner trengs for å beregne DFT'en ved bruk av radix-2 FFT-algoritmen?
   - Hvor mange multiplikasjoner og addisjoner trengs for å beregne en 1025-punkts DFT direkte?