TTT4280 Sensorer og instrumentering
Øving 2 – Fast Fourier Transform

Fouriertransformasjonen er en av de viktigste matematiske verktøyene for å analysere et målt signal, fordi den forteller hvilke og hvor mye av hver frekvens man finner i signalet. Dette er særs nyttig, siden det vi mennesker oppfatter som lyd er periodiske svingninger i lufttrykk, mens lys er periodiske svinginger i det elektromagnetiske feltet. Det er også derfor man lærer om dette i blant annet TTT4120 Digital signalbehandling. Det kommer ikke til å bli noe mindre nyttig i dette faget; denne øvingen fungerer som en oppfriskning i hvordan man gjør Fourieranalyse i Python med `NumPy` eller `SciPy`.

Oppgaver

1. Et tenkt sinussignal har amplituden 1 V og frekvensen 100 Hz, og måles med samplingstid $\Delta t = 0.2$ ms.

   Lag sinussignalet $x(t)$ bestående av en vektor på 900 punkter langs tidsaksen, der avstanden Δt mellom hvert målepunkt er 0.2 millisekund, og amplituden og frekvensen er som beskrevet over. Plott de første 200 datapunktene. Husk å få på riktig benevnelse og skalering på tidsakse og y-akse. Hva blir samplingsfrekvensen f_S og Nyquist-grensen (i Hz) i dette tilfellet?

2. Vi skal nå se på frekvensspekteret til signalet ved hjelp av Fouriertransformasjonen. Vanligvis kjenner vi ikke til den underliggende funksjonen som produserer signalet; vi kjenner kun til signalverdien ved de gitte måletidspunktene. Dermed må vi nøye oss med en diskret fouriertransformasjon, der man vanligvis foretrekker å bruke Fast Fourier Transform-algoritmen (FFT). Her skal vi se på noen av de praktiske konsekvensene for frekvensspekteret man får fra å ta en diskret fouriertransformasjon.

   (a) Skriv et kodestykke som tar FFT av signalet med 1024 datapunkter ($N_{\text{FFT}} = 1024$). Lag en frekvensakse av samme lengde (dvs 1024 punkter), fra 0 Hz til 5 kHz. Husk å regne ut intervallet Δf (frekvens-skritt) mellom datapunktene langs frekvensaksen.

   (b) Dataene som kommer ut fra FFT, kaller vi $X(f)$. Kvadratet av dette kalles ofte et *periodogram*, og dette brukes som et estimat av effekttetthetsspekteret $S_{XX}(f)$. Plott hele effekttetthetsspekteret $S_{XX}(f) = X^*(f)X(f) = |X(f)|^2$ opp til 5 kHz slik at en ser speil-frekvensen (alias) i spekteret. Hvilken frekvens opptrer denne på?

   (c) Zoom inn på spekteret slik at du plotter frekvenser mellom 0 Hz og 200 Hz, og kontroller at maksverdien til spekteret havner på riktig sted langs frekvensaksen.

   (d) Plott effekttetthetsspekteret $S_{XX}(f)$ i desibel ved å ta funksjonen $20\log_{10}(|X(f)|)$ og normaliser amplituden til spekteret slik at maksimalverdien blir 0 dB. (y-aksen kan settes til å dekke området mellom -80 dB og 10 dB). Korrekt benevnelse på y-aksen blir nå «`Relativ effekt, [dB]`».

3. Dette plottet inneholder *all* informasjon som målepunktene kan gi oss om signalet, men det kan se ganske taggete ut. Vi ønsker å gjøre det frekvensspekteret enklere å lese ved å gjøre grafen glattere". Det kan man få til ved såkalt *zero-padding*, som legger til 0-ere i slutten av signalvektoren, slik at tidssignalet nå får en total lengde på 4096 punkter.

   (a) Ta en ny FFT med lengde $N_{\text{FFT}} = 4096$ og plott spekteret. NB: Husk å sørge for at frekvensskrittene Δf i frekvensaksen blir riktige når N_{FFT} økes.

   (b) Beskriv hva som skjer med plottet til spekteret. Øk gjerne N_{FFT} til 2×4096 for å se effekten tydeligere.

4. Zero-padding er bare én måte å gjøre signalet enklere å lese på. Egentlig er det bare et eksempel på det å bruke en *vindusfunksjon* i tidsdomenet. [1] Vi skal nå gjøre det samme, men med et Hanning-vindu som har samme lengde som selve sinussignalet. Signalet multipliseres punkt for punkt med verdiene i denne vindusfunksjonen før du bruker FFT.

   Plott frekvensspekteret med og uten Hanning-vindu i samme plottet. Hva skjer med nivået på sidelobene og bredden på hovedloben når en bruker vindusfunksjon?

5. Signalet som du har brukt hittil er et såkalt reellt signal. Erstatt dette signalet med en komplekst harmonisk signal $x(t) = \exp(-j\omega t)$, der $\omega = 2\pi f$.

   Legg ved koden for alle deloppgavene i besvarelsen til oppgave 5.

   (a) Plott spekteret for dette signalet opp til samplingsfrekvensen f_S.

   (b) I virkeligheten vil frekvensene som ligger mellom $f_S/2$ og f_S representere negative frekvenser. (Dvs. at data fra denne delen av spekteret egentlig tilhører negativ frekvensakse). Rearranger frekvensaksen og frekvensplottet vha funksjonen `numpy.fft.fftshift` slik at 0 Hz blir liggende midt langs x-aksen. (Sjekk at du får riktig indeks på frekvensaksen ved f.eks å sette $f = 0$ på det komplekse sinussignalet. Da skal maksverdien til spektret havne på 0 Hz).

   (c) Endre fortegnet på det komplekse signalet slik at en får $x(t) = \exp(+j\omega t)$. Hvor havner maksverdien til spekteret i dette tilfellet?

[1] Det at zero-padding fungerer som en vindusfunksjon høres kanskje rart ut med det første, men det forklarer hvorfor frekvensspekteret får sitt karakteristiske zero-paddin-utseende. Å zero-padde en funksjon er matematisk ekvivalent med å utvide signalet du har målt i løpet av tiden T til det nye antallet punkter, før du multipliserer det med en boksfunksjon $U(t) = \begin{cases} 1, \, t \in [0, T] \\ 0, \, t \notin [0, T] \end{cases}$, slik at halen til det utvidede signalet $x(t) \cdot U(t)$ blir 0. Fourierspekteret $\mathcal{F}\{x(t)U(t)\} = \mathcal{F}\{x(t)\} * \mathcal{F}\{U(t)\}$, der $*$ indikerer *konvolusjonen* mellom de to funksjonene. Det viser seg at $\mathcal{F}\{U(t)\} = \frac{\sin(\pi T f)}{\pi T f}$, som du sikkert kjenner fra før. Hvis ikke, kan du plotte funksjonen og se om du ser noen likheter mellom den og frekvensspekteret et zero-paddet-signal.