

Tid og frekvens

Dag Kristian Dysthe and Anja Røyne

Fysisk institutt, UiO

(Dated: January 16, 2012)

Målet i denne oppgaven er å få et bevisst forhold til hvordan man måler tid. Vi vil gå fra helt grunnleggende til mer avanserte metoder. Et viktig poeng er også hvordan man beregner “usikkerhet” og å finne ut av hvordan man skal minimere denne. Sist, men ikke minst, skal dere trene på å lage en rapport og å føre en labjournal.

I. BAKGRUNN

Måling av tid baserer seg på periodiske hendelser. Årssyklusen og soluret er de eldste og best kjente tidsmålerne. Hvor nøyaktig man kan måle tid kommer an på hvor regulære periodene er (f.eks. jordrotasjonen) og hvor fint man kan dele opp perioden i underenheter (f.eks. antall streker på sirkelen i et solur). Det vi kaller en klokke i dag er et instrument med noe som svinger (f.eks. pendelen i et pendelur, en elektronisk svingekrets, eller lysbølgen fra en veldefinert kvantemekanisk overgang) og noe som teller antall svingninger. Dagens beste tidsstandard har en nøyaktighet på under ett sekund på 30 millioner år. Det er den mest nøyaktige standarden for noen fysisk enhet vi kan måle på. Se f.eks. <http://tf.nist.gov/general/museum/847history.htm>.

Vi skal i denne øvelsen bare bruke “helt vanlige” oscillatorer. Vi skal se på hva som ligger i begrepene “nøyaktighet” og “presisjon” og hvordan man som fysiker forholder seg til at alt man måler har en “usikkerhet”. Kapittel 2, 3 og 10 i Squires er pensum og nyttig for å løse denne øvingen.

II. LITT TEORI

A. Normalfordelingen

Normalfordelingen er også kjent som Gaussfordelingen, og er beskrevet i kapittel 3 i Squires. Det er såpass vanlig å støte på normalfordelinger når det er usikkerhet i målinger, at dette er stoff du bør ha satt deg inn i før øvingen. Matlab har en del funksjoner for normalfordelinger som du kan få bruk for, og et utvalg av dem presenteres her. I det følgende antas det at X er en vektor som inneholder N måleresultater.

- `mean(X)`
Som navnet på funksjonen tilsier, gir dette middelverdien av samtlige målinger i X .
- `std(X)` eller `std(X,1)`
Dette gir standardavviket til X , forskjellen på de to er at førstnevnte skalerer standardavviket med $N - 1$, mens sistnevnte skalerer standardavviket med N . Begge versjoner av standardavviket kan finnes i forskjellige lærebøker, Squires bruker førstnevnte.

- `normrnd(μ , σ , N, 1)`
Denne funksjonen returnerer en vektor som er N lang med tilfeldige tall hentet fra en normalfordeling med middelerverdi μ og standardavvik σ .

- `normpdf(Y, μ , σ)`
Denne funksjonen returnerer en sannsynlighetsfordeling gitt ved en normalfordeling med middelevrider μ og standardavvik σ , evaluert i punktene valgt i vektoren Y .

- `hist(X,M)`
Med denne funksjonen får du plottet et histogram av dataene i X fordelt i M søyler. Merk at både for mange og for få søyler gir dårlige histogrammer. Om du ønsker et histogram hvor søylenes totale areal er normalisert til én, kan det ordnes med følgende kode:

```
%Sett parametre
N = 10000; %Antall datapunkt
mu = 0;    %Middelerverdi
sigma = 1; %Standardavvik
```

```
%Lag syntetiske data
X = normrnd(mu, sigma, N, 1);
```

```
%Lag histogram med 100 søyler
[n,xout] = hist(X,100);
```

```
%Finn totalt areal for søylene
area = sum(n(2:end).*diff(xout));
```

```
%Plot normert histogram
bar(xout,n/area);
%Kunne også plottet med
%plot(xout,n/area,'.'), som vil gi punkter.
```

```
%Sammenlikning med perfekt normalfordeling
hold on;
plot(xout,normpdf(xout,mu,sigma),'r');
```

På siste linje plottes en analytisk normalfordeling for valgt middelevrider og standardavvik, dette kan være en grei måte å vurdere om måledataene er normalfordelte.

- `normplot(X)`
Denne funksjonen viser grafisk hvor nær normal-

fordelt et datasett er. Hvis datasettet er normalfordelt, blir alle datapunktene liggende på en rett linje, mens eventuelle avvik fra en normalfordeling vil gi kurvatur i plottet.

Det er også mulig å plote en kumulativ fordeling for datasettet:

```
%Sett parametre
N = 10000; %Antall datapunkt
mu = 0;    %Middelverdi
sigma = 1; %Standardavvik

%Lag syntetiske data
X = normrnd(mu, sigma, N, 1);

%Sorter datasettet i stigende
%rekkefølge
XX = sort(X);
M = (1:N)/N;

%Plot kumulativ fordeling
plot(XX,M);

%Plot for normalfordeling
%Analytisk løsning for integralet av
%normalfordelingen
Erf=(1+erf((XX-mu)/(sqrt(2)*sigma)))/2;
hold on;
plot(XX,Erf,'k:');
```

III. LABORATORIEØVING

A. Timeglass og pendel

Opp til fire grupper deler ett timeglass. Hver gruppe har en pendel.

- Mål perioden til timeglasset i antall pendelsvingninger.
- Hva vil dere angi som nøyaktigheten i målingene deres?
- Mål avstanden fra opphengspunktet til massesenteret til pendelen og beregn den teoretiske svingetiden for pendelen i sekund. Dermed kan dere beregne timeglassets periode i sekunder.
- Hva er de viktigste kildene til usikkerhet?
- Hvilke er tilfeldige og hvilke er systematiske feil?
- Sett tall på (estimert eller ved gjentatte målinger) alle feilkildene.
- Dere har nå “kalibrert” timeglasset. Hvilken presisjon og nøyaktighet har dette timeglasset som tidsmåler?

B. Pendel og stoppeklokke

- Mål perioden til pendelen med en stoppeklokke. Bruk mellomtidsfunksjonen til å få fortløpende, gjentatte målinger.
- Er mellomtidene normalfordelt?
- Hva er nøyaktighet, presisjon og hovedfeilkilder i målingen av svingetiden til pendelen?
- Sammenlign målingen med den teoretisk beregnede svingetiden til pendelen.
- Mål timeglasset noen ganger med stoppeklokken mens dere klargjør til oppgave III C. Hva vil dere nå si er nøyaktigheten til timeglasset som tidsmåler? Kommenter presisjonen dere anga i forrige spørsmål.

C. Pendel, fotodiode og 20MHz klokke

Nå skal dere bruke en fotodiode til å måle når pendelen passerer et visst punkt i banen sin. Fotodioden består av en lysdiode som sender ut IR-lys og en lysfølsom diode som gir ut 5 Volt når den mottar reflektert lys fra lysdioden og 0 Volt når den ikke mottar lys. Fotodioden krever 15 Volt drivspenning. Positiv drivspenning går til rød inngang, utsignalet er på hvit inngang og jord er på svart.

Last ned “*svingeperiode.m*” til lab-PC-en, start Matlab og åpne “*svingeperiode.m*” i editoren. Akvisisjonsboksen, NI USB-6211, er styrt av PC-en via en USB-port. Den har en intern svingekrets (20 MHz) som holder takten på når den foretar seg noe. Boksen kan gjøre en rekke ting, men i dag skal dere bare bruke en analoginngang, det vil si to kontaktpunkter der boksen måler spenningen og omformer det til et digitalt tall som er proporsjonalt med spenningen. Den kan måle spenninger opp til 250 000 ganger i sekundet. Det uttrykkes som at samplingraten er maksimalt 250 kHz. Siden taktholderen er på 20 MHz skal variasjonen i perioden mellom to samplinger være mindre enn $5 \cdot 10^{-8}$ s.

Bruk loddet som reflektor eller klistre en bit aluminiumstape på loddet som reflektor. Koble et multimeter på utgangen fra fotodioden for å teste hvor det er best å plassere fotodioden i forhold til reflektoren på pendelen. Koble så måleledningene fra akvisisjonsboksen (ledningene skal være koblet til AI0 (inngang 15) og AI GND (inngang 28)) til utgangen på fotodioden. Nå kan dere sette igang pendelen og kjøre scriptet “*svingeperiode.m*” for å måle svingetiden til pendelen.

- Hvordan er svingetiden nå sammenlignet med målingen med stoppeklokke?
- Betyr det noe
 - hvilken samplingsfrekvens dere bruker?

- hvor i pendelbanen fotodioden plasseres?
- hvor stort utslaget av pendelen er?
- Er mellomtidene normalfordelt?
- Hva er nøyaktighet, presisjon og hovedfeilkilder i målingen av svingetiden til pendelen?

IV. UTSTYRSLISTE

- Timeglass
- Pendel (lodd, tråd, festeanordning)
- Stoppeklokke
- Fotodiode (svart liten boks)
- Måleledninger
- Meterstokk
- Spenningsforsyning
- Akvisisjonsboks NI USB-6211
- PC
- Umbrakonøkkel
- Sett med små skruttrekkere
- Blank tape