TTT4280 Sensorer og instrumentering
Øving 1 - Måleusikkerhet

Oppgaver

1. Vi har laget et målesystem som måler temperaturen i et rom en gang hver tiende minutt. Oppvarmingssystemet er laget slik at temperaturen burde holdes på 20 grader. Temperaturmålingene for de første 20 minutene er disse:

$$T_{\text{maalt},1} = \begin{matrix} 20.6, & 20.4, & 20.4, & 20.6, & 20.4, & 20.8, & 20.5, & 20.5, & 20.5, & 20.4, \\ 20.5, & 20.5, & 20.5, & 20.5, & 20.4, & 20.4, & 20.4, & 20.5, & 20.3, & 20.6 \end{matrix}$$

   (a) Gi et estimat for standardavviket for disse målingene.

   Løsningsforslag. Numerisk får vi

$$m_T = \frac{1}{20}\sum_{i=1}^{20} T_i = 20.49 \text{ grader},$$

$$s_T = \sqrt{\frac{1}{19}\sum_{i=1}^{20}(T_i - m_T)^2} = 0.11 \text{ grader}.$$

   (b) Gi et 95% konfidensintervall for middeltemperaturen i rommet.

   Løsningsforslag. Vi kan bruke Student's t-fordeling for å beregne et konfidensintervall for forventingsverdien, μ_T:

$$\mu_T = m_T \pm t_p \frac{s_T}{\sqrt{n}},$$

   hvor m_T er sampelmiddelverdien, s_T er estimatet av standardavviket, og n er antallet målepunkter. I deloppgave (a) beregnet vi allerede et estimat for standardavviket og middelverdien:

$$m_T = 20.49 \text{ grader},$$

$$s_T = 0.11 \text{ grader}.$$

   I tabell for t-fordelingen, med 95%, dobbelsidet fordeling, og 19 frihetsgrader, finner vi $t_p \approx 2.1$ og da er altså

$$\mu_T = 20.49 \pm 0.05 \text{ grader}$$

   (c) Kan du si innen hvilket intervall som neste målepunkt med 95% sannsynlighet vil ligge?

Hint: For en normalfordeling ligger 95% innenfor ±1.96 teoretiske standardavvik fra forventingsverdien. Vi kjenner ikke forventingsverdien eller det teoretiske standardavviket, men kun estimat av disse fra våre 20 målinger. Da blir *prediksjonsintervallet* for neste målepunkt.

$$T \in m_T \pm t_p s_T \sqrt{1 + \frac{1}{n}}$$

Løsningsforslag. For en normalfordeling ligger 95% nesten presis innenfor ±2 teoretiske standardavvik fra forventingsverdien. Vi kjenner ikke forventingsverdien eller det teoretiske standardavviket, men kun estimat av disse fra våre 20 målinger. Da blir *prediksjonsintervallet*

$$T \in m_T \pm t_p s_T \sqrt{1 + \frac{1}{20}} = 20.49 \pm 0.23 \text{ grader}$$

2. Nå har vi utviklet et mye enklere (billigere) målesystem for å måle temperaturen. Oppvarmingssystemet er detsamme, så temperaturen i rommet burde være densamme. Vi måler igjen hver tiende minutt:

$$T_{\text{maalt},2} = \begin{matrix} 20.4, & 20.4, & 20.4, & 20.2, & 20.4, & 20.3, & 20.4, & 20.5, & 20.4, & 20.4, \\ 20.4, & 20.4, & 20.1, & 20.3, & 20.3, & 20.2, & 20.3, & 20.2, & 20.3, & 20.3 \end{matrix}$$

(a) Igjen, gi et 95% konfidensintervall for middeltemperaturen i rommet.

Løsningsforslag. Samme metode igjen gir:

$$\mu_T \in 20.33 \pm 2.1 \frac{0.098}{4.47} \approx 20.33 \pm 0.05$$

(b) Er de to konfidensintervallene for middelverdien (i oppgave 1 og oppgave 2) overlappende? Hvis ja, så betyr det at vi ikke ser noen statistisk signifikant forskjell på (middelverdien for) de to målesystemene.

Løsningsforslag. Intervallet i oppgave 1 var 20.44 - 20.54 grader, og i oppgave 2 var det 20.28 - 20.38 grader. Disse konfidensintervallene overlapper ikke og altså gir de to målesystemene en statistisk signifikant forskjell i middelverdi. De to målesystemene har da forskjellige systematiske feil, f eks grunnet kalibreringsfeil.

(c) Hvis du kun hadde hatt de 10 første målingene, ville du ha dratt densamme sluttsatsen?

Løsningsforslag. Vi gjør om alle beregninger (og må bruke $t_p = 2.3$) og får intervallene for oppgave 1: 20.51 $\pm 2.3 \cdot \frac{0.13}{3.16} \approx 20.51 \pm 0.09$ grader, og for oppgave 2: 20.38 $\pm 2.3 \cdot \frac{0.079}{3.16} \approx 20.38 \pm 0.06$ grader. Disse to intervallene overlapper akkurat og da kan vi ikke trekke sluttsatsen, med 95% sikkerhet, at de to måleseriene gir forskjellig middelverdi.

3. Vi har kjøpt motstand med resistansen $R = 1$ kΩ, med presisjonen 1%, og vi kan anta at resistansverdien har en *uniform* fordeling ±1%.

(a) Hva er maksimal verdi, minimal verdi, og standardavvik for motstandsverdiene, i Ω?

Løsningsforslag. Maksimal verdi er $R_{max} = 1010\Omega$, minimal verdi er $R_{min} = 990\Omega$. Standardavviket for en uniformfordeling, mellom R_{max} og R_{min} er (se formelsamlingen):

$$\sigma_R = \frac{R_{max} - R_{min}}{\sqrt{12}} = \frac{20}{3.46} \approx 5.8\Omega$$

(b) Hva er det *relative* standardavviket, i %?

Løsningsforslag. Relativt standardavvik er $\frac{\sigma_R}{\mu_R} = \frac{5.8}{1000} = 0.0058 = 0.58\%$.

(c) Nå kjøper vi motstand med resistansen $R = 500\ \Omega$, og seriekobler to for å få resistansen 1 kΩ. Hva blir sannsynlighetsfordelingen for resistansverdien for den her seriekoblingen av to motstand? Du kan tegne, eller forklare i ord, eller formulere matematisk.

Løsningsforslag. Når vi summerer to tilfeldig valgte motstandsverdier så blir sannsynlighetsfordelingen for summen = en konvolusjon av to uniforme sannsynlighetsfordelinger. En slik konvolusjon gir en triangelformet funksjon med toppen på summen av middelverdiene, i vårt fall er det da en topp ved 1kΩ. Laveste mulige verdien (dvs starten på den triangelformete fordelingsfunksjonen) er $(1000 - 10)\Omega = 990\Omega$, og den høyeste mulige verdien er $(1000 + 10) = 1010\Omega$.

(d) Hva blir det relative standardavviket, i %, for denne seriekoblingen av to motstand? Du kan beregne standardavviket numerisk, f eks ved å bruke funksjonen `numpy.std` i Python.

Hint: Du kan beregne standardavviket for et eneste motstand ved å simulere i Python (så kan du også sammenligne med den teoretiske verdien for en uniform fordeling!):

Listing 1: Lag et stort antall verdier med uniform fordeling, med ønsket middelverdi, og spredning.

```
import numpy as np
delta_r = 10
r_values = 1000 + 2*delta_r * (np.random.rand(int(1e5)) - 0.5)
relative_std = np.std(r_values)/np.mean(r_values)
```

`int()` må brukes da `numpy.random.rand()` kun tar inn heltall, mens `1e5` vil tolkes som et flyttall.

Løsningsforslag. Vi løser dette numerisk i Python:

```
import numpy as np
nsamples = 10000
R1 = 500
dR1 = 0.01*R1
Rvec1 = R1 + 2*dR1 * (np.random.rand(nsamples, 2) - 0.5)
Rvecsum = np.sum(Rvec1, axis=1)
print(np.mean(Rvecsum))
print(np.std(Rvecsum)/np.mean(Rvecsum))
```

Da får vi relative standardavviket 0.41%. Vi får altså et noe mindre gjennomsnittlig feil ved å seriekoble to motstand enn å bruke en motstand, hvis alle har samme presisjon.