



AALBORG UNIVERSITY
DENMARK

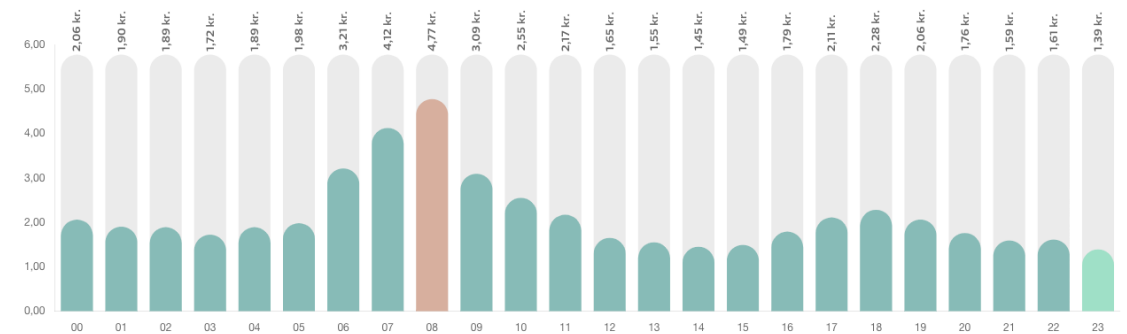
GRPROG Workshop II

Energibesparende styring af kølerum

Martin V. Vejling, Tobias Kallehauge, Jimmy J. Nielsen
{mvv,tkal,jjn}@es.aau.dk
Department of Electronic Systems
Aalborg University
9220 Aalborg, Denmark

Motivation

- En restaurant opbevarer madvarer i kølerum ved 5°C
- Hver gang døren åbnes af personale stiger temperaturen
- En termostat sørger for at starte kompressoren, så temperaturen holdes omkring 5°C [on/off]
- De stigende og kraftigt svingende timepriser for el gør at restauranten overvejer en intelligent styring
- Q: Hvor meget kan spares?



Modellering

- Diskret tid:
 - 5 minuts interval, $N = 30 \cdot 24 \cdot 12 = 8640$ samples
 - 4 mulige tilstande i hvert minut: dør [åben/lukket], kompressor [on/off]
- Temperaturudvikling udregnes iterativt fra differensligningen:

$$T[n] = T[n - 1] + (C_1(T_{\text{rum}} - T[n - 1]) + C_2(T_{\text{komp}} - T[n - 1])) \Delta t,$$

$$T_{\text{rum}} = 20^\circ$$

$$T_{\text{komp}} = -5^\circ$$

$$\Delta t = 300 \text{ s}$$

$$C_1 = \begin{cases} 5 \cdot 10^{-7} \text{ s}^{-1} & \text{hvis døren er lukket} \\ 3 \cdot 10^{-5} \text{ s}^{-1} & \text{hvis døren er åben} \end{cases},$$

$$C_2 = \begin{cases} 0 \text{ s}^{-1} & \text{hvis kompressoren er slukket} \\ 8 \cdot 10^{-6} \text{ s}^{-1} & \text{hvis kompressoren er tændt} \end{cases}.$$

Termostat

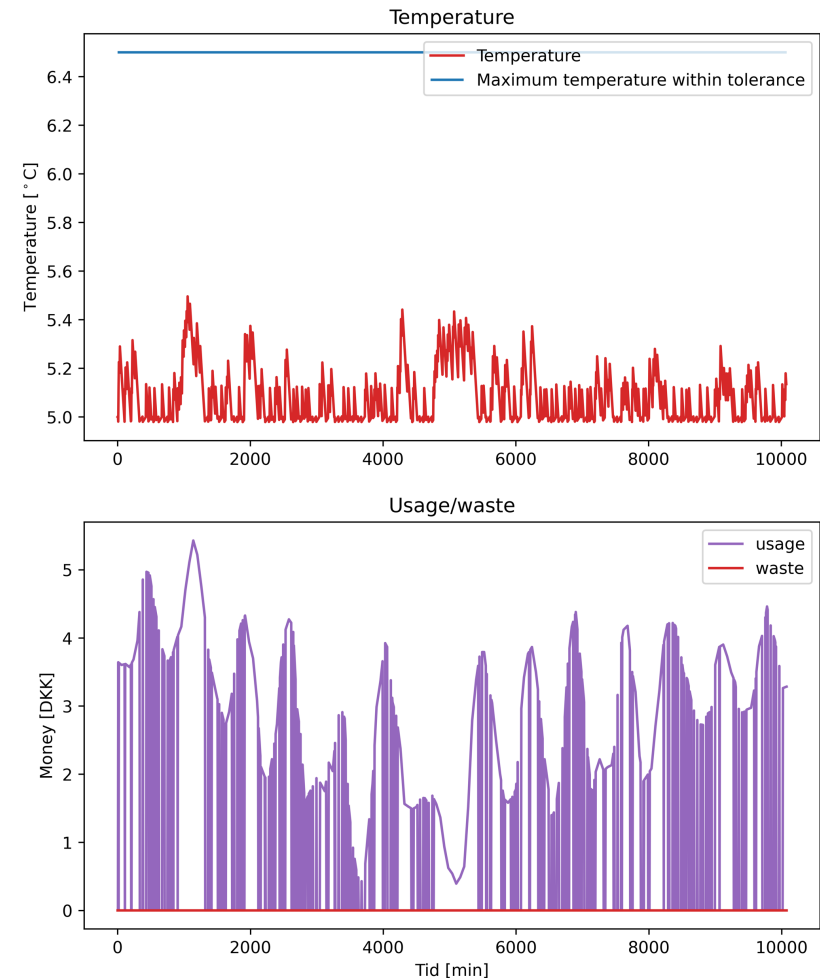
- Simpel styring ud fra temperatur:

$$\text{termostat}[n] = \begin{cases} \text{tændt} & \text{hvis } T[n] > T_{\text{mål}} \\ \text{slukket} & \text{hvis } T[n] \leq T_{\text{mål}} \end{cases}$$

- Vi antager:

$$T_{\text{mål}} = 5^{\circ}$$

- Eksempel på temperatur og pris over 1 uge (høje spikes skyldes åben dør).



Dør

- Personale henter og bringer madvarer i kølerummet.
- Vi antager at døren i hvert tidsinterval kan være enten åben eller lukket.
- For et vilkårligt tidsinterval er:

$P = \text{sandsynlighed for at døren er åben} = 10\%$

Elpris og forbrug

- El afregnes til markedspris i “kr per kilo Watt time” [kr/kWh]

$$\text{forbrug}[n] = \begin{cases} \text{elpris}[n] \cdot W_{\text{komp}} \text{ kr.} & \text{hvis kompressoren er tændt} \\ 0 \text{ kr.} & \text{hvis kompressoren er slukket} \end{cases}$$

- Vi antager at kompressorens forbrug er: $W_{\text{komp}} = 1 \text{ kWh}$ for hvert 5. minut.
- Elpriser for september 2022 hentes fra datasættet `elpris.csv` i moodle.

Tab af madvarer ved temperaturudsving

- For høj temperatur fordærver madvarerne
- For lav temperatur kan give kulde/frostskader
- Omkostninger til madspild beregnes som:

$$\text{madpris}(T) = \begin{cases} 4.39e^{-0.49T} \text{ kr.} & \text{hvis } T < 3.5^\circ \\ 0 \text{ kr.} & \text{hvis } 3.5^\circ \leq T < 6.5^\circ \\ 0.11e^{0.31T} \text{ kr.} & \text{hvis } T \geq 6.5^\circ \end{cases}$$

Total udgift

- Summen af omkostninger til elforbrug og madspild:

$$\begin{aligned} \text{udgift} &= (\text{forbrug}[0] + \text{madpris}(T[0])) + \cdots + (\text{forbrug}[N - 1] + \text{madpris}(T[N - 1])) \\ &= \sum_{n=0}^{N-1} (\text{forbrug}[n] + \text{madpris}(T[n])) \end{aligned}$$

- Bemærk at en realisering af september måned (med tilfældige åbninger af dør) kan have forskellig udgift fra én simulering til den næste.
- Derfor benytter vi Monte-Carlo simulering til at finde gennemsnitlig udgift over M simuleringer:

$$\text{gennemsnitlig udgift} = \frac{1}{M} \sum_{m=1}^M \text{udgift}_m$$

Problemet

- I workshoppen stilles 2 opgaver:
 1. Simuler kølerummet, og udregn den gennemsnitlige udgift for september 2022 når den simple termostat bruges. Bestem ud fra dette om restauranten kan overholde et månedligt budget på 12.000 kr.
 2. Eksperimentér nu med en intelligent termostat:
 - Elpriserne kendes som regel på forhånd, og det antages derfor at den intelligente termostat har adgang til elpriserne i fremtiden (dvs. I kan opsætte regler der tager disse med i betragtning for [on/off]).
 - Det kan dog ikke forudsiges hvornår døren åbnes.
 - Bestem herefter om restauranten ved hjælp af den intelligente termostat kan overholde sit budget (det er IKKE et krav at den kan).
 - Vi forventer ikke at I bruger mange timer på denne opgave, men prøv at være lidt kreative og eksperimentér. I må meget gerne arbejde sammen med jeres gruppe til dette! Hvad er den laveste totale udgift I kan opnå?

Krav til løsning

- Koden til løsningen afleveres i moodle workshop. Følgende krav er givet:
 1. Programmet skal kunne løse opgave 1. og 2. herover.
 2. Strukturen til programmet afhænger af studieretningen:
 1. Computerteknologi: Programmet skrives objektorienteret med minimum to klasser:
 - Undgå således kode udenfor klasserne samt globale variable.
 - Skriv også en passende docstring til hver metode.
 2. Ingeniørvidenskab: Programmet skal opbygges fortrinvis af Python funktioner
 - undgå således kode uden for funktioner og globale variable så vidt muligt.
 - Til hver funktion skrives en passende docstring.
 3. Til minimum 3 af funktionerne/metoderne skrives en doctest, der kan bruges med Pythons doctest modul.
 - Implementer automatisk doctest af disse ved brug af `if __name__ == "__main__":` syntaksen.
 4. Programmet skal opdeles i minimum 2 moduler (Python scripts).
 - Til hvert modul skrives en passende docstring i starten af filen.

Dokumentation

- Sammen med kode afleveres lille rapport (pdf, max 5 sider eller 10 slides):
 1. En beskrivelse af den intelligente termostat, du har implementeret og den gennemsnitlige udgift den giver.
 2. Minimum 3 plots, der illustrerer problemet og dets løsning samt sammenligning med simpel termostat. Her er et par idéer til plots:
 - Tidsserieplot af temperaturen i kølerummet i løbet af en uge (det kan være svært at forholde sig til en hel måned i et plot).
 - Tidsserieplot, der viser både udgiften per 5. minut til el og madspild hen over en uge.
 - Histogram over udgift per 5. minut i september.
 3. Et kort resume af resultaterne fra de to opgaver: Hvad er den gennemsnitlige månedlige udgift ved de to typer termostater og kan restauranten overholde sit budget?

Opgaver

- Følgende opgaver er givet:
 - Forstå hvordan kølerummet og udgifter simuleres og forstå problemet.
 - Lav en plan for hvordan programmet skal opbygges. Hvilke funktioner/metoder og moduler skal implementeres?
 - Implementér programmet i Python jævnfør ovenstående krav og lav dokumentationen.
 - Aflever program og dokumentation senest d. 28. november kl. 23:59.
 - Lav peer review af de andre studerendes løsninger. Deadline 30. november.
- Hver studerende afleverer sin egen kode og rapport, men I må gerne diskutere problemer og løsninger indbyrdes.