

# GFV

## Labøvelse 3 - Automatisk temperatur regulator

Aarhus universitet - Softwareteknologi

Simon Phi Dang	Lucas Friis-Hansen	Shynthavi Prithviraj
201705957	201811527	201807198

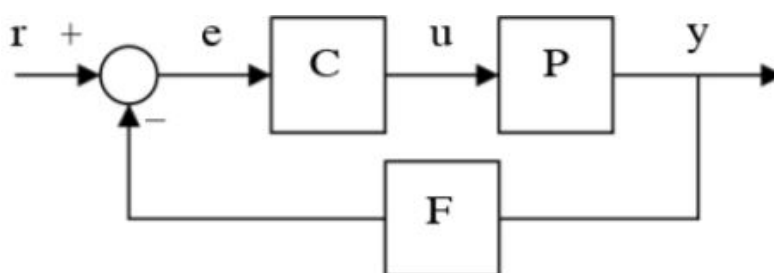
10-11-2020

# Indhold

<b>1</b>	<b>Introduktion</b>	<b>2</b>
<b>2</b>	<b>Temperature controller</b>	<b>3</b>
2.1	Formål . . . . .	3
2.2	Design og implementering . . . . .	3
2.3	Realisering og Resultat . . . . .	4
2.3.1	Observeringer ved test . . . . .	7
2.4	Diskussion . . . . .	11
2.5	Konklusion . . . . .	13

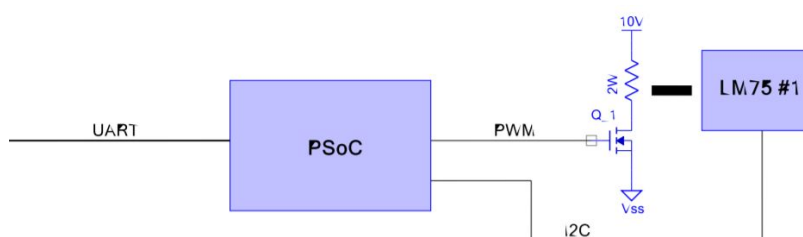
# 1 Introduktion

Dette eksperiments hovedformål er at forstå de basiske principper af lukkede loop kontrol systemer illustreret af en PID kontroller. Der anvendes en software implementering af PID kontroller for at nå og opretholde en konstant temperatur. Derudover vil påvirkningen af at vælge forskellige koefficienter for P og I parterne af kontrolleren blive undersøgt. Nedenstående på figur 1 vises et block diagram af det lukkede loop kontrol system. Diagrammet viser at outputtet af systemet er  $y$ , som er givet gennem en sensormåling  $F$  til sammenligning med referenceværdien  $r$ . Kontrolløren  $C$  tager error  $e$  mellem referencen og outputtet til at ændre input  $u$  til systemet under kontrol  $P$ . (OMSKRIV)



Figur 1: Blokdigram af det lukkede loop kontrol system [1]

På figur 2 vises en overordnet skitse af eksperimentet. Kredsløbet skal forsynes med 10V og aktuatoren implementeres som en 2W strømresistor, der forbindes til MOSFET PCB. Billedet er dermed også med til at vise, hvad implementeringen skal indeholde (UART, I2C og PWM). Ved hjælp af I2C kan PSoC og LM75 snakke sammen. PWM bruges til at kontrollere strømmen, der forsvinder i strøm resistoren. Og UART til at få outputtet vist.



Figur 2: Diagram/skitse af det lukkede loop kontrol system [1]

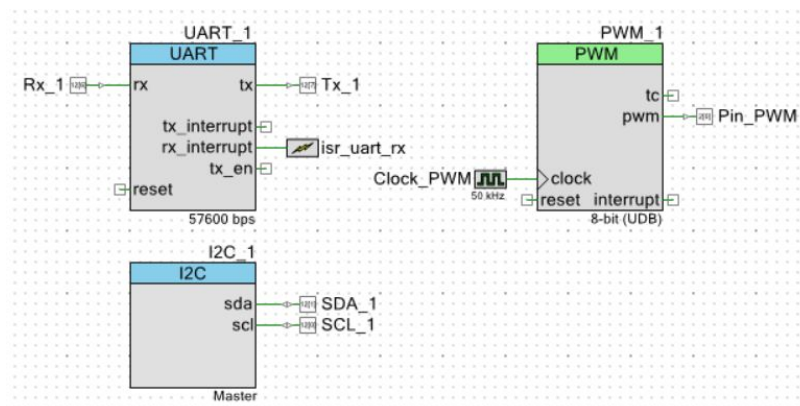
## 2 Temperature controller

### 2.1 Formål

I denne øvelse skal temperaturen aflæses fra en temperaturmåler(LM75) på samme måde som for Eksperimentet for Communication busses. For at kontrollere strømmen, der forsvinder i resistoren, anvendes et PWM-signal, som forbindes til en gate på MOSFET. Dette minder meget om måden, motorhastigheden blev kontrolleret på i en tidligere opgave.

### 2.2 Design og implementering

Der oprettes en PSoC Creator, hvor implementeringen for I2C fra den tidligere opgave indsættes. Programmet indeholder heri et endless loop, som aflæser temperaturen. For at implementere PID kontrolleren benyttes implementeringen, der er givet på Blackboard. På figur 3 vises topdesignet for eksperimentet bestående af en UART, I2C og PWM.



Figur 3: Topdesign af Eksperiment

Softwaren for PID kontroller skal opnå og opretholde en konstant temperatur. Koefficienterne for kontrolleren får værdierne:  $K_p=2$ ,  $K_i=1/30$  og  $K_d=0$ . Programmet skal heraf også indeholde, at setpoint kan blive ændret mellem 30 grader og 50 grader. Ved at have disse setpoints, kan man sammenligne udførelsen af kontrolleren med forskellige kontroller koefficienter. Derfor gøres følgende:

```

void handleByteReceived(uint8_t ByteReceived)
{
    switch(ByteReceived)
    {
        case '3':
        {
            UART_1_PutString("30 set\n");
            setPoint = 30; //Setpoint sættes til 30 grader
            PIDControl_changeSetPoint(30);
        }
        break;
        case '5':
        {
            UART_1_PutString("50 set\n");
            setPoint = 50; //Setpoint sættes til 50 grader
            PIDControl_changeSetPoint(50);
        }
        break;
        default:
        {
        }
        break;
    }
}

```

Figur 4: Kode af setpoint

På figur 4 vises, at når der trykkes '3' på tastaturet ændres setpoint til 30 grader, og ligeledes trykkes '5' for at ændre setpoint til 50 grader. Resten af implementeringen af koden kan ses i den vedlagte kode.

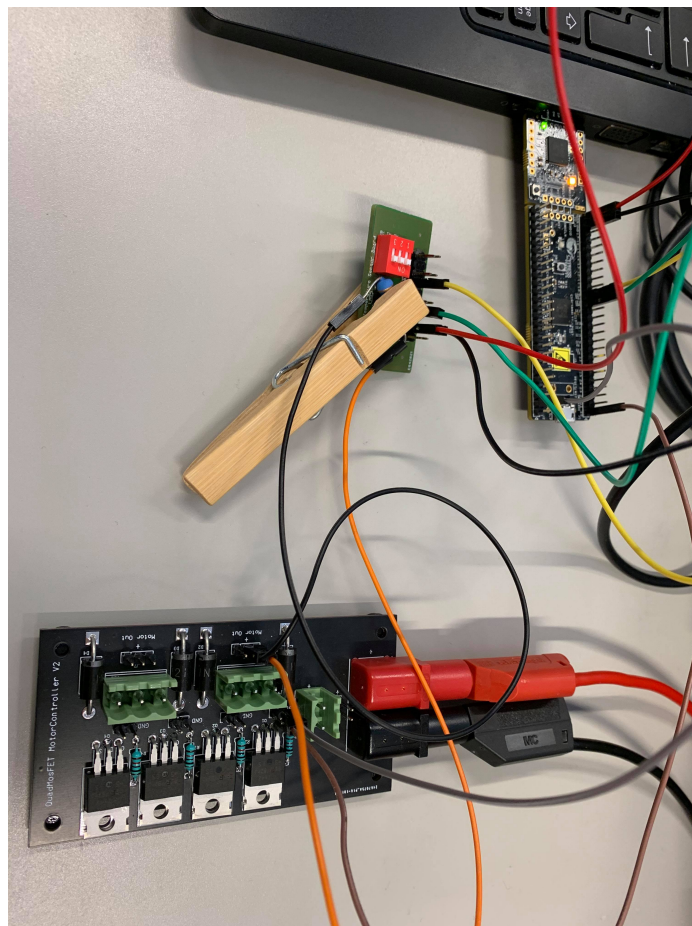
## 2.3 Realisering og Resultat

Komponenter, der er blevet anvendt:

- PSoC
- LM75 print
- MOSFET motorcontroller
- Tøjkleme
- Strømresistor

Figur 5 vises opstillingen for eksperimentet. PSoC'en sættes til PC'en, og forbindes til LM75 og MOSFET motorcontrolleren. Der bruges en strømforsyning til at forsyne resistoren 10 Volt via

MOSFET controlleren. Resistoren lægges på LM75 og sættes fast med en tøjklamme. Der sørges for at resistoren kun rører LM75.



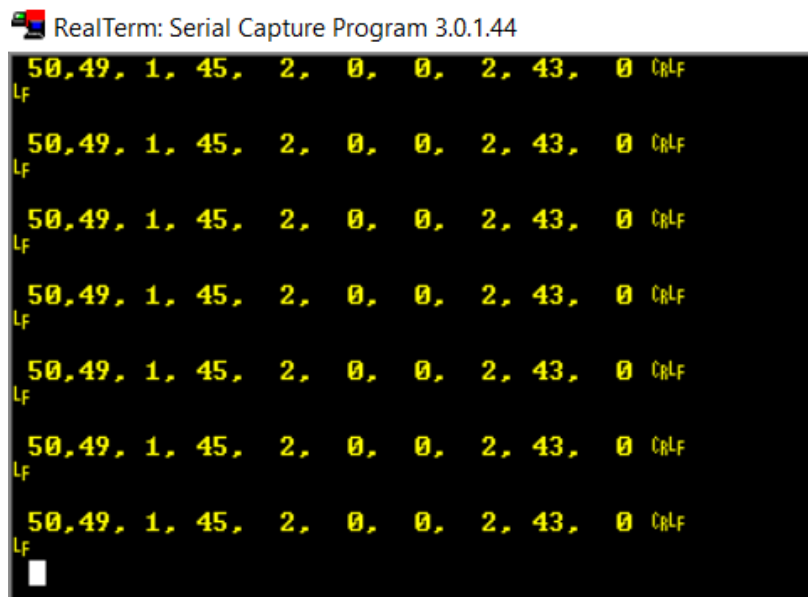
Figur 5: Opstilling af Eksperiment

Tabel 1 viser, hvad de forskellige ledninger bruges til:

<b>Sort</b>	[ <b>Ground</b> ]: Ground mellem PSoC og LM75
<b>Rød</b>	[ <b>VCC</b> ]:VCC 5V til I2C
<b>Gul</b>	[ <b>SCL</b> ]: I2C SCL
<b>Grøn</b>	[ <b>SDA</b> ]: I2C SDA
<b>Orange</b>	[ <b>PWM</b> ]: Forbindelse til power resistor, der giver PWM
<b>Sort</b>	[ <b>Ground</b> ]: Ground mellem MOSFET og LM75
<b>Grå</b>	[ <b>Ground</b> ]: Ground til MOSFET
<b>Brun</b>	[ <b>PWM</b> ]:PWM signal fra PSoC til MOSFET

Tabel 1: Farvekode til Eksperiment

Kredsløbet er forbundet som ønsket og systemet er nu klar til blive testet. Programmet eksekveres og RealTerm anvendes for at se den forventede og målte temperatur.



```

RealTerm: Serial Capture Program 3.0.1.44
50,49, 1, 45, 2, 0, 0, 2, 43, 0 CnLf
50,49, 1, 45, 2, 0, 0, 2, 43, 0 CnLf
50,49, 1, 45, 2, 0, 0, 2, 43, 0 CnLf
50,49, 1, 45, 2, 0, 0, 2, 43, 0 CnLf
50,49, 1, 45, 2, 0, 0, 2, 43, 0 CnLf
50,49, 1, 45, 2, 0, 0, 2, 43, 0 CnLf
50,49, 1, 45, 2, 0, 0, 2, 43, 0 CnLf

```

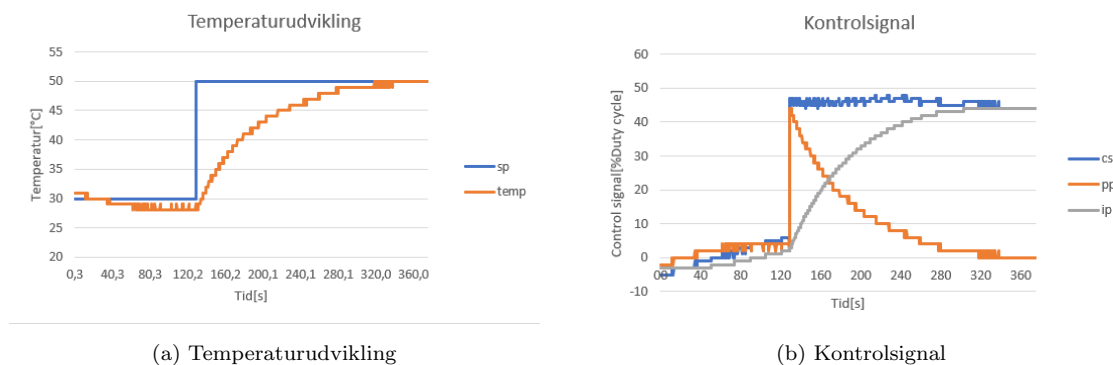
Figur 6: Konsol vindue med outputdata

På figur 6 ses konsolvinduet med en masse tal. Disse tal består af setpoint, temperatur, error, kontrolsignal,  $K_p$ ,  $K_i$ ,  $K_d$ , proportionalpart, integralpart og den afledte part. Her kan det ses, at setpoint i dette tilfælde er sat til 50,  $K_p=2$ ,  $K_i=0$  og  $K_d=0$ .

Der laves nu målinger med forskellige starting points. Rådata gemmes via RealTerms 'Capture' funktion, som kan ses i testdata mappe vedlagt i bilag.

### 2.3.1 Observeringer ved test

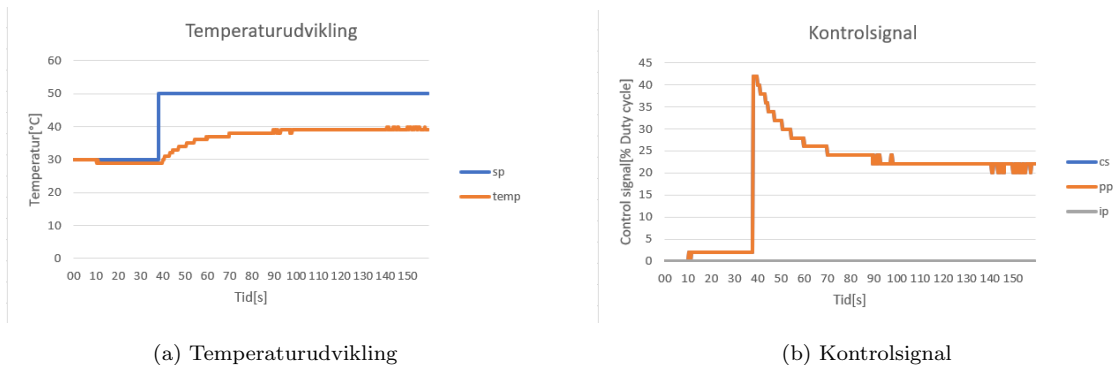
Rådata overføres til Excel, så det kan blive plottet. Som et udgangspunkt på målingerne starter  $k_p$ ,  $k_i$  og  $k_d$  værdierne til at være sat som kan ses på figur 7. På plottet ses, at når setpointet går fra 30 grader til 50 grader begynder den transiente respons. Der kan observeres på temperaturudviklingen, at temperaturen stiger med en jævn hastighed og når steady-state respons efter næsten 4 minutter (360s) ved 50 grader. På kontrolsignal-plottet ses, at proportionaldelen bidrager først ved skiftning af setpointet. Efter noget tid stiger integraledelen og proportionaldelen falder.



Figur 7: Måling 1 -  $K_p=2$ ,  $K_i=1/30$ ,  $K_d=0$

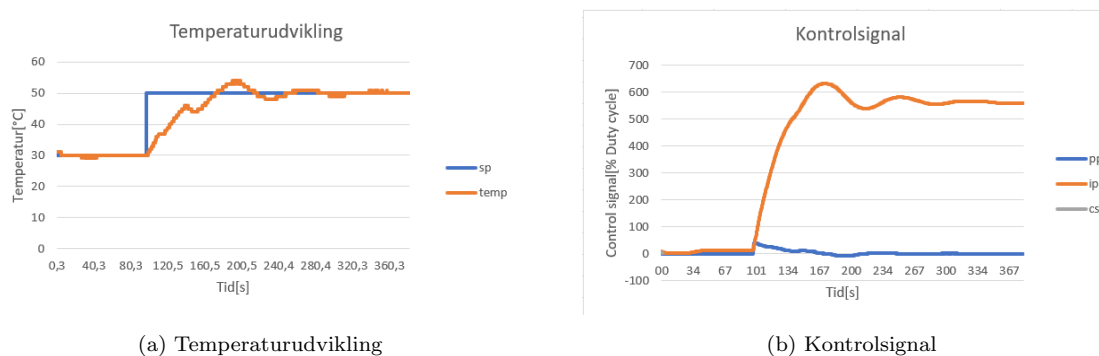
Ved ændring af integraldelen ( $K_i$ ) fra  $1/30$  til 0, kan det ses, at der opstår et permanent offset fra setpointet. Temperaturen når aldrig op på de 50 grader, som kan ses på figur 8. Dette bliver også afspejlet i kontrolsignal-plottet, hvor proportionaldelen stopper med at stige efter 40 sekunder. Herefter falder pp og når en steady-state efter 80 sekunder.





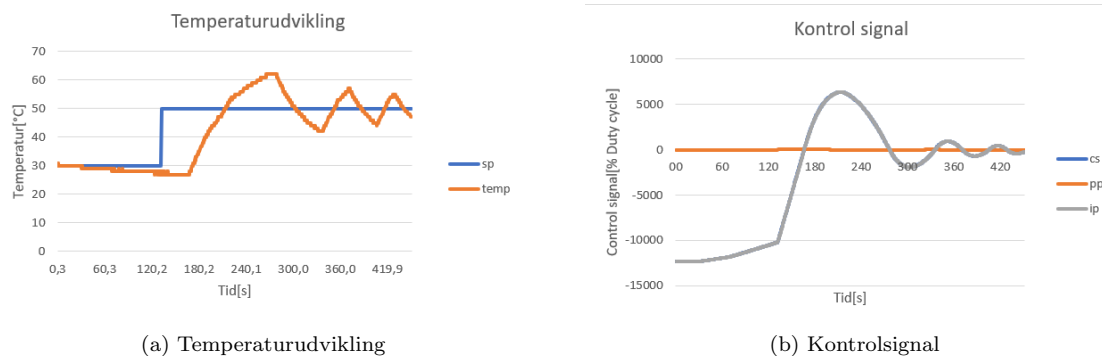
Figur 8: Måling 2 -  $K_p=2$ ,  $K_i=0$ ,  $K_d=0$

Integraldelen( $K_i$ ) ændres igen fra 0 til 1. Her ses det, at temperaturstigningen forekommer relativt hurtigt. Dog resulterer det i en overshoot, hvor den kommer over de 50 grader, men den stabiliserer sig forholdsvis hurtigt igen.



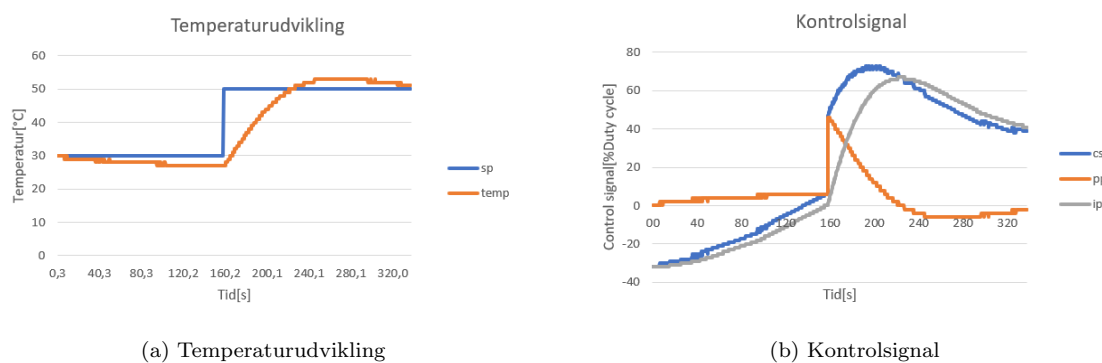
Figur 9: Måling 3 -  $K_p=2$ ,  $K_i=1$ ,  $K_d=0$

Ved at skrue helt op for integraldelen ved at sætte det til 13, set på figur 10, blev der opbygget meget negativt control signal(set i (b)), inden setpoint ændres (ses i (a)). Setpoint går fra 30 grader til 50 grader efter 120 sekunder og integraldelen begynder herefter at stige og blive positiv. Efter dette udjævnes kontrolsignalet og når setpoint efter noget tid. Igen sker der en meget høj overshoot og store svingninger, men som sagt udjævnes den.



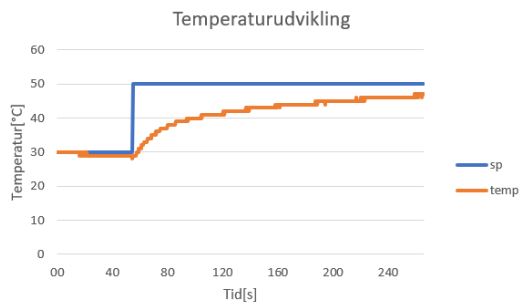
Figur 10: Måling 4 -  $K_p=2$ ,  $K_i=13$ ,  $K_d=0$

$K_i$  værdien ændres nu fra 13 til  $1/10$ . I forhold til figur 7 kan det ses på figur 11, at temperaturen stiger og når de 50 grader hurtigere, men med et lille overshoot. Dette afspejles også i kontrolsignal-plottet til højre, hvor det falder igen efter der opnås setpunktet. På figur 7 er den mere stabil lige efter den når setpunktet.

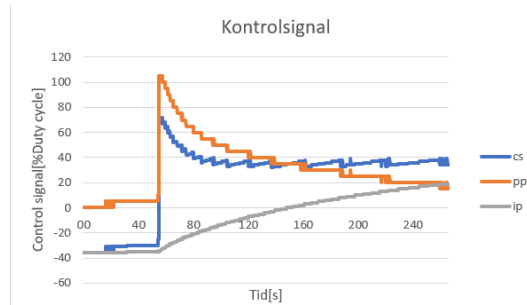


Figur 11: Måling 5 -  $K_p=2$ ,  $K_i=1/10$ ,  $K_d=0$

Figur 12 viser hvorledes det tager længere tid for temperaturen at nå setpunktet, da proportional-delen( $K_p$ ) øges.



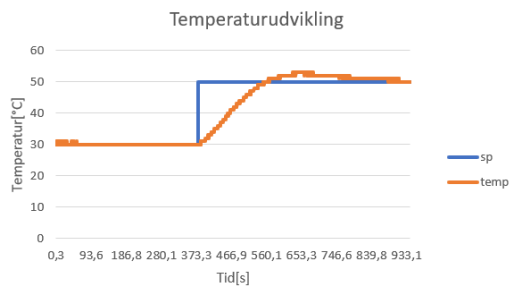
(a) Temperaturudvikling



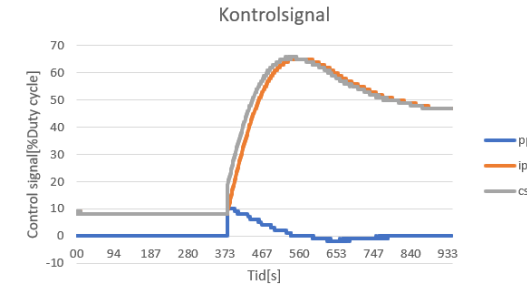
(b) Kontrolsignal

Figur 12: Måling 6 -  $K_p=5$ ,  $K_i=1/30$ ,  $K_d=0$

På figur 13 er der sat en meget lav  $K_p$  værdi, hvilket gør at proportionaldelen ikke har en stor påvirkning, hvilket også kan ses på kontrolsignalet (b). Integraldelen dikterer, hvordan control signalet ser ud.



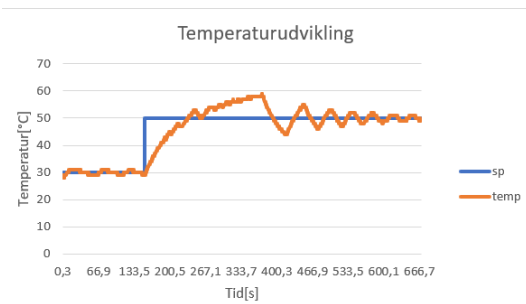
(a) Temperaturudvikling



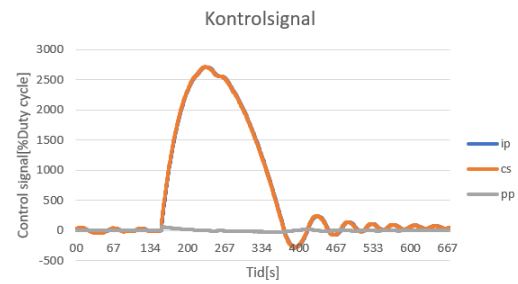
(b) Kontrolsignal

Figur 13: Måling 7 -  $K_p=0.5$ ,  $K_i=1/30$ ,  $K_d=0$

Ved at både øge  $K_p$  værdien og  $K_i$  værdien kan der ses, at  $K_p$  værdien ikke er høj nok til at have en stor betydning for control signalet. Integraldelen og kontrolsignalet ligger ovenover hinanden. Temperaturudvikling viser også, at der er en hurtig temperaturstigning til setpointet, hvor man så også får overshoot. Udsvingningerne jævner sig dog ud og rammer setpoint.



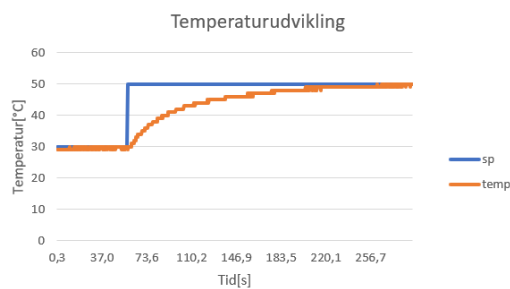
(a) Temperaturudvikling



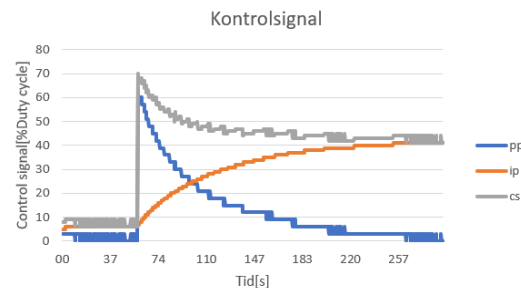
(b) Kontrolsignal

Figur 14: Måling 8 -  $K_p=3$ ,  $K_i=4$ ,  $K_d=1$

Til sidst ændres  $K_p$  værdien til at være 3, som det ses på figur 15. Der opnås en hurtigere stigning til setpunktet, men stadigvæk vedligeholdes den stabile kurve opad. Der forekommer ikke en overshoot og årsagen er, at der skal være en stor nok P del, som kan håndtere I delen, så der ikke overshootes.



(a) Temperaturudvikling



(b) Kontrolsignal

Figur 15: Måling 9 -  $K_p=3$ ,  $K_i=1/30$ ,  $K_d=0$

## 2.4 Diskussion

**How does changing the coefficients for the P and I parts affect the controller? (use a systematic approach and only change one coefficient at the time, start by changing  $K_p$ )**

Der er kommet frem til, at når I delen ændres, kan det ses, hvorledes det påvirker PID controlleren. Ved at have meget høje I værdier var der en hurtig stigning i temperaturen. Dog opstår problemet med at temperaturen vil overshoot og overkorrigere. Ud fra at kigge på forskellige værdi ændringer

var værdien  $K_i=1/30$  god til at have en stabil stigning. Det kunne godt ændres til  $1/10$  eller  $1/20$ , da det kun forårsager en lille overshoot ved test. Dette ses på figur 11. Ved at ændre på P delen kan det ses, hvordan proportionaldelen starter hurtigt med at påvirke, hvor den så falder i takt med at temperaturen stiger op til setpoint. Derudover viste det sig, at når integraldelen er meget tæt på eller større end proportionaldelen, vil proportionaldelen have en meget lille effekt på det endelige kontrolsignal.

### **Explain the effects using plots of the transient response with different coefficients**

Formålet med at anvende plots til dokumentation af resultaterne, er for at illustrere udviklingen af temperaturen ved forskellige koefficientværdier. Dette er med til at give et overblik over hvordan temperaturen ændres i forhold til tiden. På den måde kan man hurtigt vurdere samt udvælge hvilken type temperaturstyring man ønsker.

### **What would be good PID coefficients for this temperature control system?**

Hvis målet er præcision uden at temperaturen ændrer sig for hurtigt, så vil det være mest oplagt at vælge koefficienter, som matcher dem i figur 7 og 15. Her kan det ses at temperaturen har en naturlig udvikling mod den ønskede temperatur. Ønsker man derimod at få temperaturen op så hurtigt som muligt, uden at tænke på om temperaturen overstiger det ønskede, så vil man vælge koefficienter, der matcher figur 14.

### **Does the sampling rate affect the performance of the controller?**

Samplings rateen påvirker integrale- og den afledte del. Når samplings raten øges fører det til en mere præcis integrale og en mere reagerende kontrol output. Altså vil sampling rate påvirke controllerens performance. Der sker dog det, at når sampling rate øges, øges støjet også i outputtet, da den afledte del ofte kan ændre sig.

### **How is integral windup avoided in the controller?**

Integrator windup opstår når der sker en stor ændring i setpoint, som gør at integralet fejler under 'opstigning', i dette tilfælde til den ønskede temperatur. Dette fænomen er også kendt som 'overshooting' og ses tydeligt på figur 10, hvor temperaturen overstiger set point med  $+10\text{ }^{\circ}\text{C}$ .

For at forhindre denne udfordring, så bør man sikre at det stigende setpoint sker i et fornuftigt tempo. Det vil altså sige, at man ikke skal fokusere på at få ændret setpoint så hurtigt som muligt,

men at man i stedet for bør sikre at præcisere stigningen ved hjælp af kontrolsignalet. Dette ses på figur 7, hvor kontrolsignalet (cs) matcher 'set point' (sp) og følger det ret nøjagtig. På den måde kan man sikre en mere naturlig temperaturudvikling, som ikke 'over shooter' den forventede temperatur.

## 2.5 Konklusion

I denne labøvelse har det været muligt at få implementeret et system, som aflæser temperaturen fra en LM75-sensor. Det er muligt at afmåle temperaturen, og derefter skrive en ny ønsket temperatur (50 C°) ved hjælp af UART. Når det nye 'set point' er registreret, så vil der blive afgivet strøm til power modstanden indtil den ønskede temperatur er nået.

## Litteratur

- [1] Labøvelsesvejledning pdf fra underviser