

# Laboratorieøvelse # 3

## Automatic Temperature Regulator

**Dato: 28/03/2019**

**Gruppe 21**

**#1**

Stud.nr.: 201404212    Navn: Rasmus Vesterheim

**#2**

Stud.nr.: 201710712    Navn: Marcus Bech

**#3**

Stud.nr.: 201710685    Navn: Martin Krøjmand

## Indholdsfortegnelse

<b>Indledning</b>	<b>2</b>
Experiment 1: Temperature controller	<b>3</b>
Teori	3
Opsætning	4
Implementering	4
Målinger	6
Diskussion og databehandling	13
Konklusion	<b>16</b>

# Indledning

Denne laboratorieøvelse går ud på at udvide I2C øvelsen med temperatursensoren fra sidst og DC motor control fra Lab #1 til at lave en temperaturregulator. I stedet for at måle på to slaveenheder, så vil denne opgave tage udgangspunkt i at kunne regulere og kontrollere temperaturen på det ene temperatur-print i stedet. Det gøres med en 2W power resistor, der styres af PWM.

## Experiment 1: Temperature controller

### Teori

Denne PID controller, vi vil implementere i denne øvelse, skal være i stand til at regulere temperaturen af en 2W power resistor. Implementeringen vil være i form af en closed loop controller med feedback, som ses på nedenstående figur:

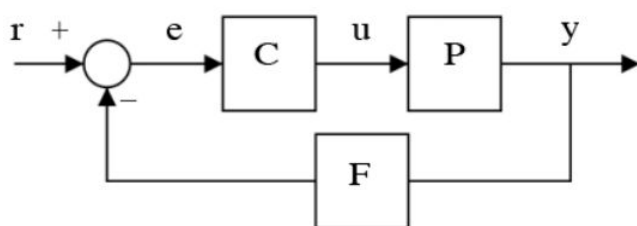


Figure 1 Skitse af closed loop controller med feedback

I denne øvelse er  $y$  lig med temperaturen af 2W modstanden, som bliver målt af en temperatursensor (LM75), der er repræsenteret ved  $F$  i ovenstående figur. Den målte temperatur bliver sammenlignet med en referenceværdi,  $r$ , som vi vil kontrollere igennem controlleren,  $C$  (PSoC). Controlleren vil reagere på forskellen imellem disse to,  $e$ , og ændre PWM-outputtet,  $u$ , afhængigt af denne forskel.

Størrelsen af  $u$  afhænger af tre led, som ses på figuren nedenunder.

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt}$$

Figure 2 Ligning for outputtet  $u(t)$  fra controlleren  $C$

Det ses, at det første led er proportionelt med fejlmargenen for den nuværende fejl; det andet led tager integralet af fejlmargenen over tid, mens det tredje led differentierer fejlmargenen over tid. Læg mærke til, at denne formel er det, som  $u(t)$  ideelt set bliver. Eftersom vi behandler fejlene digitalt via samples, så vil integralet altså i praksis være summen af arealet af rektangler med 1 sample længde bred.

$K_p$ -leddet vil altid give én værdi baseret på fejlmargenen for den nuværende fejl. Der tages ikke højde for udviklingen af de forrige samples.

Ki-leddet vil stige støt med udviklingen af tiden så længe, den målte værdi er forskellig fra referenceværdien (setpoint).

Til sidst vil Kd-leddet se på udviklingen af fejlmargenen, da den kigger på hældningen af e for det nuværende tidspunkt (baseret på forrige og nuværende sample). Kd-leddet vil ikke blive brugt i denne øvelse, så Kd er sat til 0 i vores implementering.

## Opsætning

På figur 3 ser vi vores test opstilling. Her har vi udvidet opstillingen fra sidst med vores LM75 temperatursensor og PSoC. Vi har altså fået udleveret vores temperaturprint, som vi bruger til at varme en 2W power modstanden op, som er sat fast til LM75 med en klemme.

Vi har monteret SDA(data) og SCL(clocken) på LM75 printet. Herudover er vcc og ground også forbundet fra PSoC til print. Output pins fra den ene Mosfet fra MOSFET printet er sat til modstanden. Inputtet af MOSFET printet er vores PWM signal hvor vi bestemmer outputtets størrelse fra PSoC'en. Printet er desuden tilsluttet en forsyningspænding på 10V fra en strømforsyning, som aflades i 2W power modstanden.

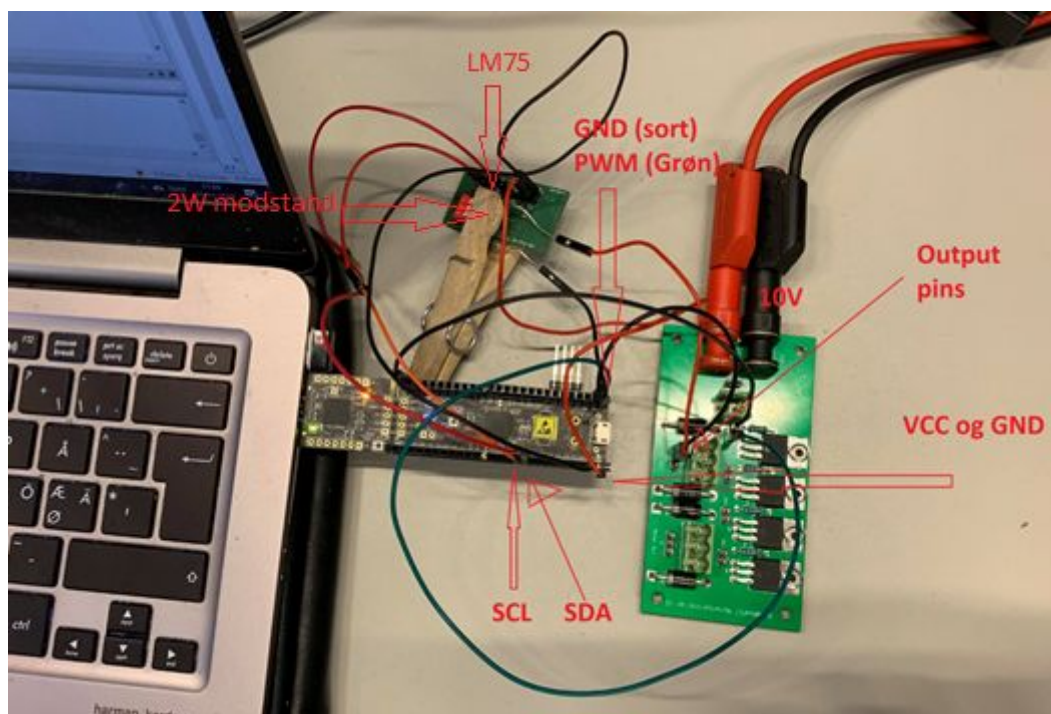


Figure 3 Opsætningen af forsøget

## Implementering

I implementeringen af denne temperaturregulator er blokkene brugt og dokumenteret førhen i de to tidligere lab-øvelser. Her er der tale om UART til at printe temperatur, koefficienter, PWM osv. ud. I2C er for at hente data fra temperatursensoren, som forårsager, at PWM ændrer sig. PWM sørger for at ændre styrken af den genererede varme fra power resistor. Disse blokke ses i figuren herunder:

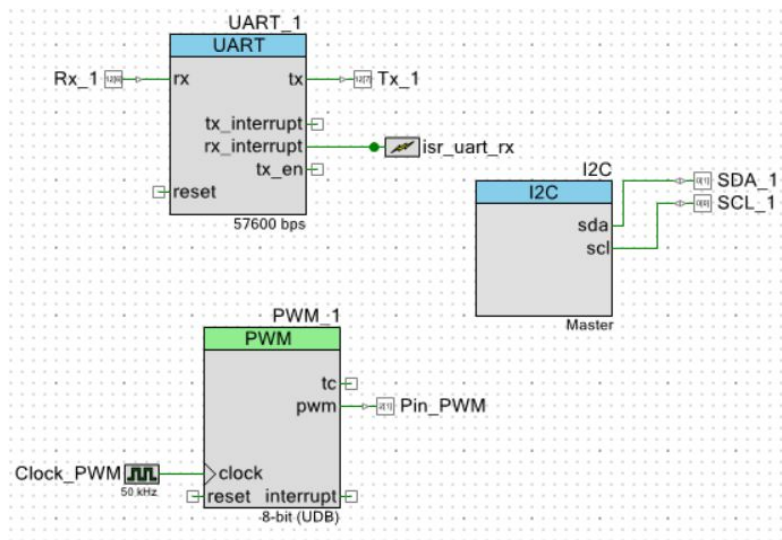


Figure 4 Billede af komponenter i PSoC projektet

Meget af koden fra reguleringen af temperaturen er taget fra Blackboard i GFV for Lab øvelse 3. Der gennemgås dog lidt af koden for, hvordan proportional leddet og integrale leddet er fundet:

```
// calculate current error
currentError = setPoint - systemOutput;

// calculate proportional part
proportional = currentError;
```

Figure 5 Udsnit af beregning for proportional-del

Proportional-leddet er ganske enkelt forskellen mellem setpoint og den temperatur, der er målt i den nuværende måling. Herefter ses integrale leddet:

```
// calculate integral part
integral = integral + (currentError * dt);

// limit the integral
if (integral > integralMax) integral = integralMax;
if (integral < integralMin) integral = integralMin;
```

Figure 6 Udsnit af beregning af integrale-del

Integralet er summen af alle arealer for rektangler under grafen for fejl (afvigelse fra setpoint). Derfor bliver integralet for de forrige samples hele tiden lagt til, når et nyt integrale værdi skal sættes. Arealet for et rektangel under grafen er produktet af bredden og højden. Bredden af sample tiden er udtrykt ved dt. Værdien af dt er derfor 0,33 (s) i vores tilfælde. Højden er den diskrete værdi for fejlen. For det nye integrale er det værdien for den nuværende fejl.

Kd leddet er ikke så relevant for denne øvelse, da den bliver sat til 0, og det er ikke meningen, at vi skal eksperimentere med denne koefficient.

```
output = proportional * Kp + integral * Ki + derivative * Kd;
*proportionalPart = proportional * Kp;
*integralPart = integral * Ki;
*derivativePart = derivative * Kd;
```

Figure 7 Udsnit af output-ligningen

Både proportional-led, integrale-led og differentieret-led bliver lagt sammen med de medfølgende faktorer, som vi bestemmer: Kp, Ki og Kd. Kd bliver dog ikke brugt i dette eksperiment.

```
previousError = currentError;
if(output>100) output = 100;
if(output<0) output = 0;
return output;
```

Figure 8 Udsnit af del med begrænsning af output

Eftersom den nuværende fejl er blevet brugt i alle 3 led, så bliver denne fejl-værdi lavet om til previousError, som kan bruges i næste sample i differentieret led (i tilfælde af, man vil sætte Kd til mere end 0).

Der tjekkes desuden også, at dette output, som der findes frem til via formlen for u(t) (PWM outputtet), er over 100 eller under 0. Hvis det er blevet til en værdi uden for dette interval, bliver værdien lavet om til henholdsvis 100 eller 0. Hele koden ses i projektmappen for temperatur PSoC project.

## Målinger

Målingerne er udskrevet til RealTerm igennem UART på PsoC, hvorfra vi har brugt capture-funktionen til at gemme udskrifterne.

Nedenunder ses to eksempler på, hvordan udskrifterne kan se ud – på figur 9 er set-point sat til 30 grader, og den målte temperatur svinger mellem 29 og 29.5 grader. På figur 10 er set-point sat til 50 grader, og den målte temperatur svinger mellem 49 og 50 grader.

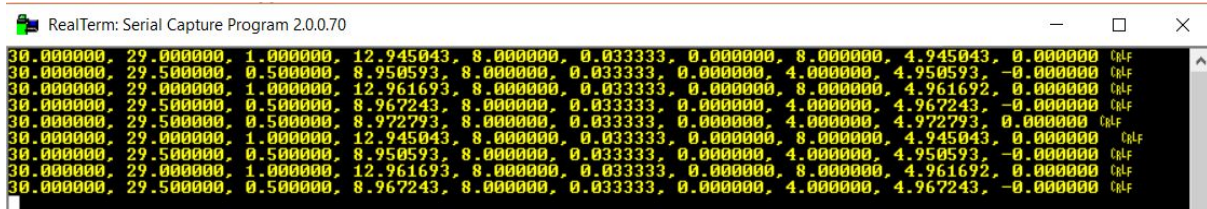


Figure 9 Billede af udskrift fra UART med set-point sat til 30 grader

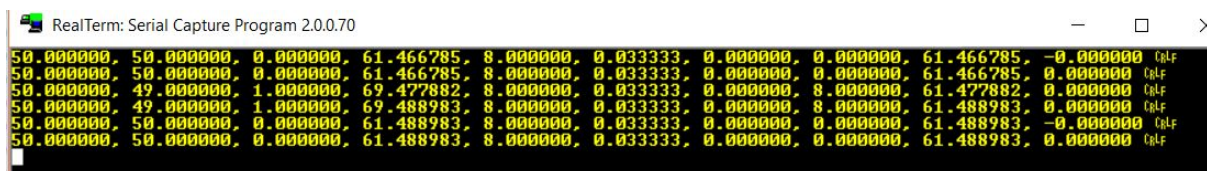


Figure 10 Billede af udskrift fra UART med set-point sat til 50 grader

Ved de første fire målinger ændrede vi kun på en af de to parametre Kp og Ki ad gangen, da vi på den måde kan se, hvad denne ændring har haft af betydning for vores temperaturregulator. Ved disse målinger brugte vi følgende parameterværdier:

Måling #	Størrelse af $k_p$	Størrelse af $k_i$
1	2	1/30
2	8	1/30
3	2	2/30
4	8	2/30

Table 1 Tabel over vores 4 første målinger

Efter de første fire målinger satte gruppen sig ned og analyserede graferne. Herefter valgte vi nogle parameterværdier, som vi forventede ville gøre, at temperaturen hurtigst muligt ville nå referencetemperaturen (setpoint).

De valgte parameterværdier kan ses på nedenstående tabel.

Måling #	Størrelse af $k_p$	Størrelse af $k_i$
5	10	3/30
6	10	5/30

Table 2 Tabel over vores målinger efter analyse af de fire første

På figur 11 til 16 ses udviklingen i den målte temperatur sammenlignet med reference temperaturen for vores målinger.

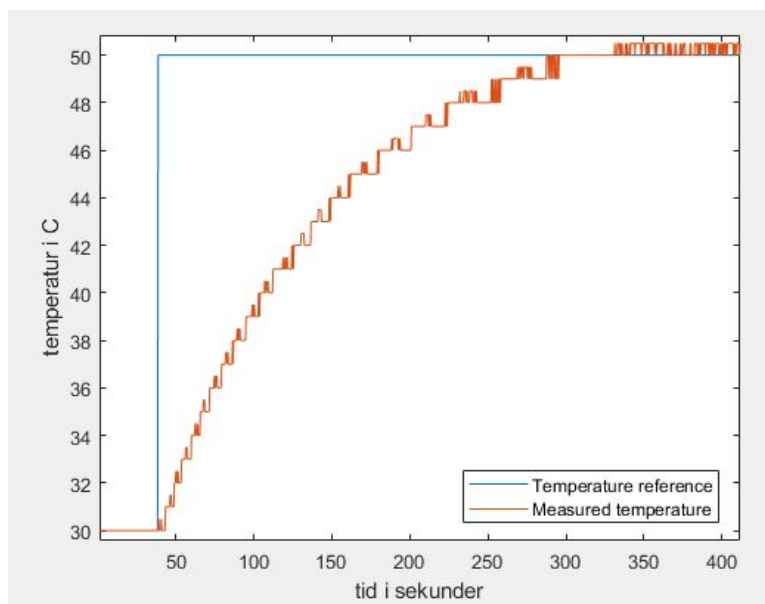


Figure 11 Billede af udviklingen af den målte temperatur med parameterværdierne  $k_p = 2$  og  $k_i = 1/30$



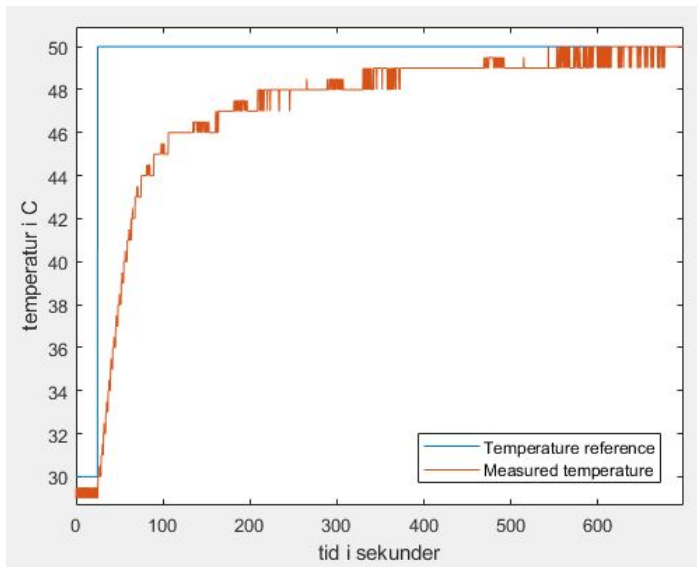


Figure 12 Billede af udviklingen af den målte temperatur med parameterværdierne  $k_p = 8$  og  $k_i = 1/30$

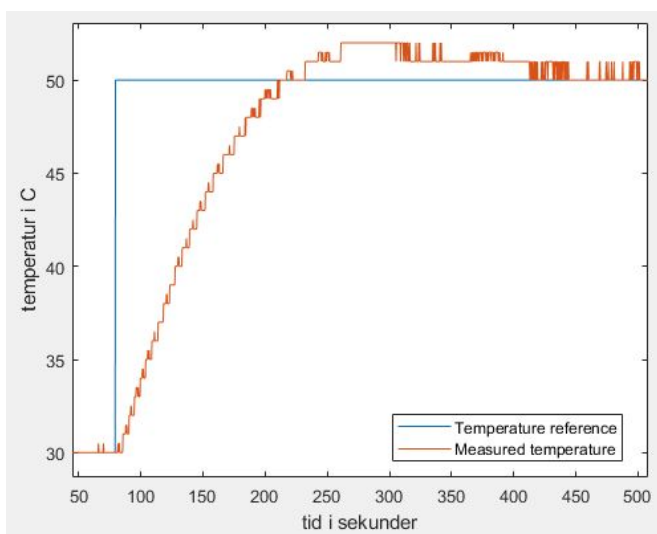


Figure 13 Billede af udviklingen af den målte temperatur med parameterværdierne  $k_p = 2$  og  $k_i = 2/30$



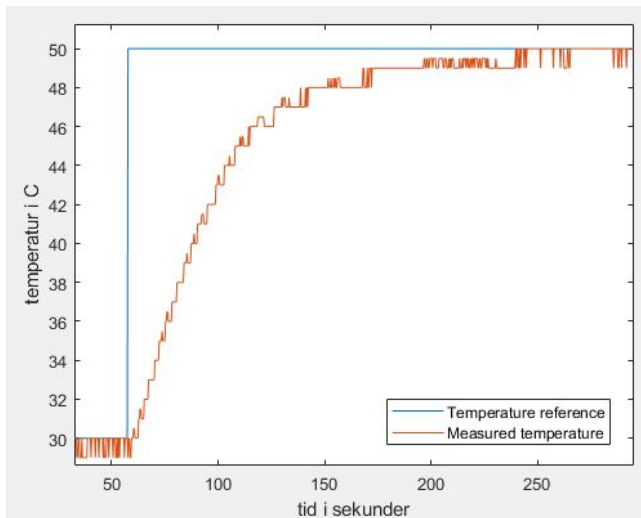


Figure 14 Billede af udviklingen af den målte temperatur med parameterværdierne  $k_p = 8$  og  $k_i = 2/30$

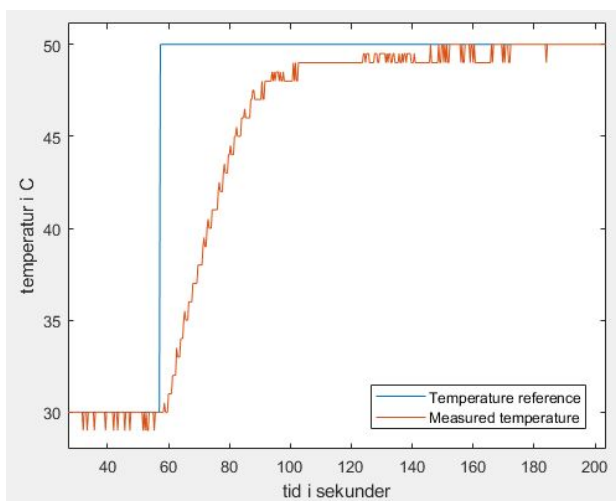


Figure 15 Billede af udviklingen af den målte temperatur med parameterværdierne  $k_p = 10$  og  $k_i = 3/30$

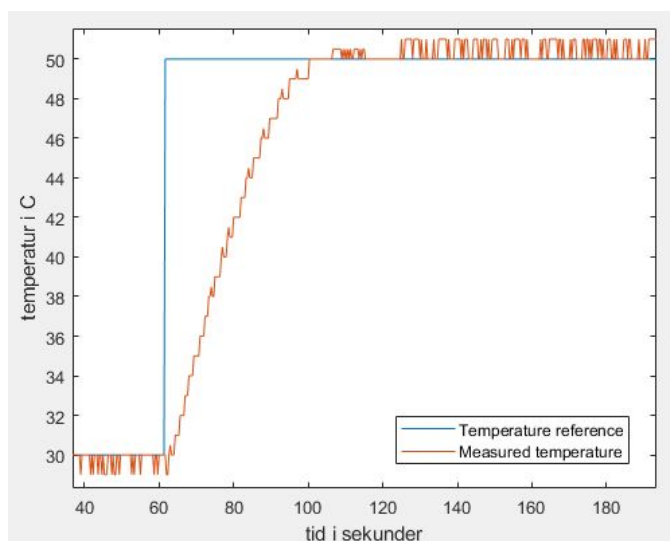


Figure 16 Billede af udviklingen af den målte temperatur med parameterværdierne  $k_p = 10$  og  $k_i = 5/30$

På de seks ovenstående figurer ses der en klar sammenhæng mellem  $k_p$  og hældningen af den målte temperatur, hvor en større  $k_p$  betyder, at temperaturen i starten af steppet stiger væsentligt mere, end når  $k_p$  er lavere.

Samtidig ses det, at en lille  $k_i$  gør, at det tager relativt lang tid at få temperaturen på præcis 50 grader, når den først er ved at nærme sig. En større  $k_i$  gør, at stigningen fra 46-50 grader går væsentligt hurtigere.

Det ses dog også på figur 16, at en for stor  $k_i$  kan gøre, at temperaturen tager lang tid om at ramme præcist 50 grader, da fejlværdien bliver ganget på en større koefficient, så denne vil have en større påvirkning på PWM-signalet.

På figur 17 til 20 nedenunder ses udviklingen af duty cycle på PWM-signalet, og påvirkningen af proportional- hhv. integral-leddet.

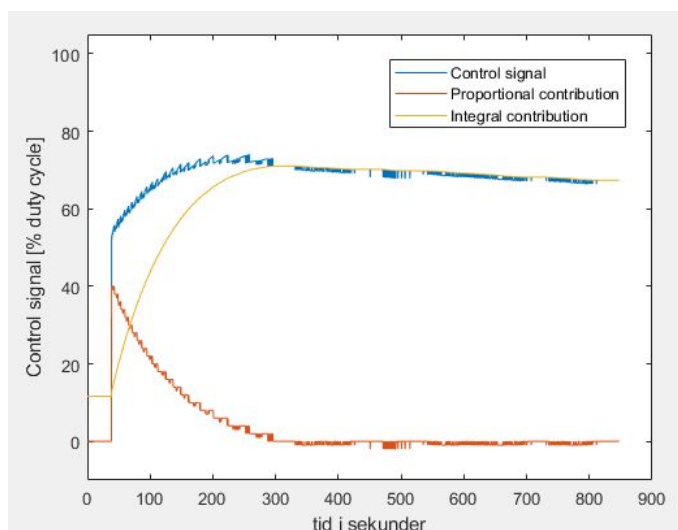


Figure 17 Billede af udviklingen af control signal, og størrelsen af påvirkning på denne af  $k_i$  og  $k_p$  med parameterværdierne  $k_p = 2$  og  $k_i = 1/30$

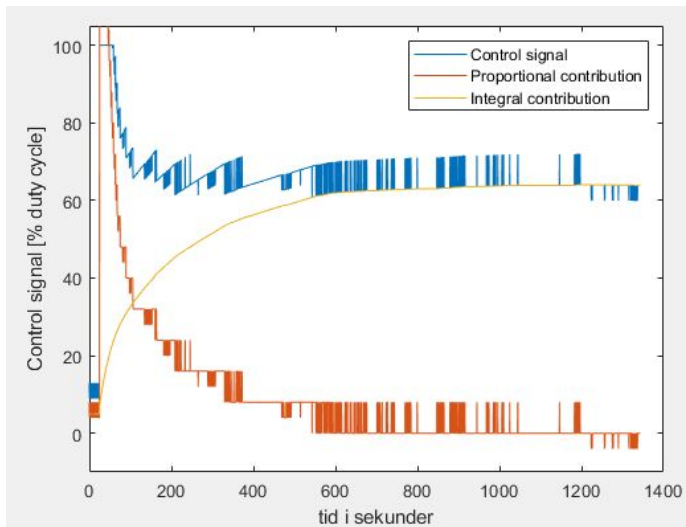


Figure 18 Billede af udviklingen af control signal, og størrelsen af påvirkning på denne af  $k_i$  og  $p_i$  med parameterværdierne  $k_p = 8$  og  $k_i = 1/30$

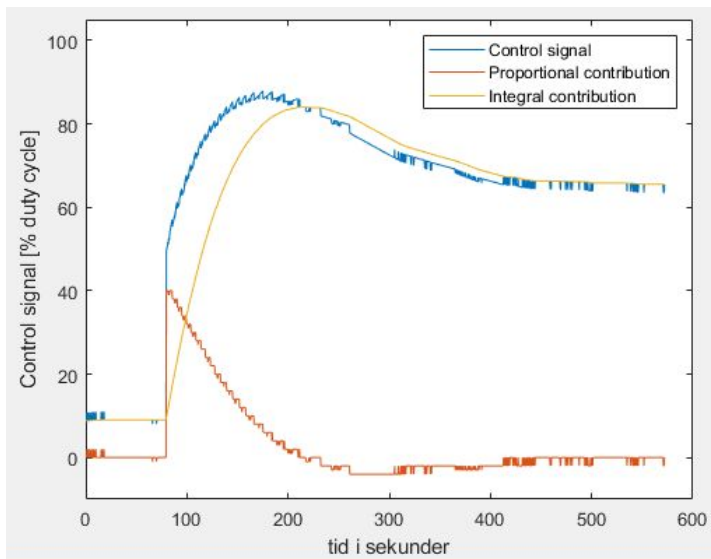


Figure 19 Billede af udviklingen af control signal, og størrelsen af påvirkning på denne af  $k_i$  og  $p_i$  med parameterværdierne  $k_p = 2$  og  $k_i = 2/30$

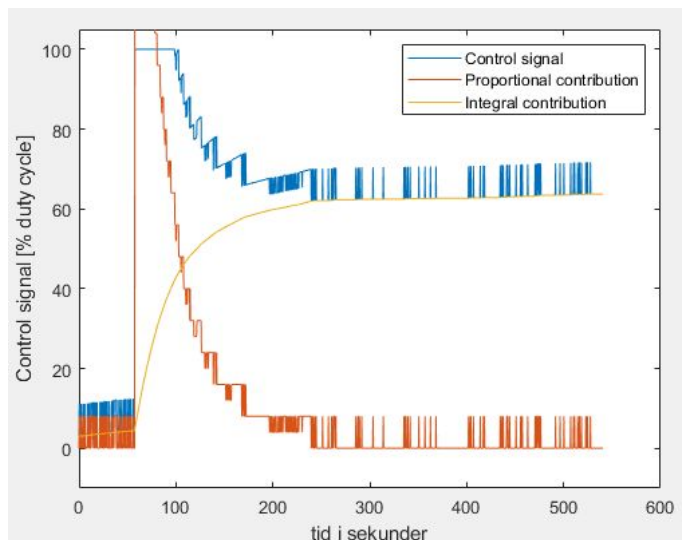


Figure 20 Billede af udviklingen af control signal, og størrelsen af påvirkning på denne af  $k_i$  og  $p_i$  med parameterværdierne  $k_p = 8$  og  $k_i = 2/30$

På de fire ovenstående figurer ses det, at en  $k_p$ -værdi på 8 gør, at den proportionale indflydelse på PWM-signalet stiger til over 100%, men da der er sat et loft på PWM-signalet, holdes dette nede. Dette betyder, som vi også så i de tidligere figurer, at hældningen af den målte temperatur er meget stejl i starten. Integrals-delen vil derefter overtage størstedelen af påvirkningen på PWM-signalet, når temperaturforskellen ikke længere er stor, og det fremgår af figurerne, at en  $k_i$ -værdi på  $2/30$  frem for  $1/30$  betyder, at denne overgang er mere flydende. På figur 18 ses, at der forekommer nogle 'hak' i control signalet, før integrals-delen bliver stor nok til at køre en tilstrækkelig høj PWM.

## Diskussion og databehandling

Ved hjælp af Real Terms capture funktion er det lykkedes også at få alle samples fra målingen med og plottet dem i Matlab, som det ses på følgende figur.

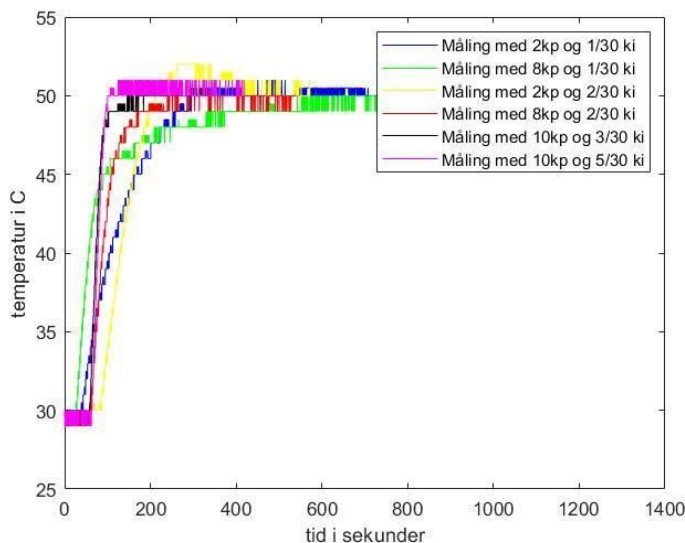


Figure 21 Samlet graf over de 6 målinger med start setpoint på 30 grader

Først og fremmest kan det ses, at det ikke er pæne kontinuerlige kurver, men derimod meget firkantet i facon. Det kommer af, at opløsningen på vores temperatursensor ikke var så god. Opløsningen var 0,5 grader. Derfor er grafen præget af firkantet steps med 0,5 grader.

Da vi lavede vores målinger, havde vi ikke overvejet, at det ville være en god idé at sørge for, at tiden, hvor målingerne er stabile på 30 grader, skulle være ens for alle målinger. Vi har derfor tilføjet ekstra punkter i starten af nogle af målingerne for at forbedre det visuelle udtryk og gøre det nemmere at sammenligne graferne. I de vedlagte målinger er disse tilføjelser ikke lavet.

Det kan ses på figur 21, at en stor kp koefficient bidrager til en meget stejl stigning af temperaturen. Derfor vil de høje kp koefficienter bidrage med en PWM på 100 % i lang tid for at få temperaturen til at stige så hurtigt som muligt. Det ses også i figuren, at den lilla og sorte kurve med kp på 10 stiger meget hurtigere end de to kurver blå og gul, der ikke har så stejl en hældning i starten.

Herimod ses Ki koefficientens effekt primært, når temperaturen nærmer sig setpointet på 50 grader og differensen i fejl bliver mindre og mindre. Her vil en lavere ki værdi sørge for at sænke stigningen af temperaturen, når det nærmer sig setpoint. Derfor undgår man at få overshoot, og heraf vil integrate windup være mindre eller slet ikke forekomme.

Til gengæld vil en højere ki koefficient sørge for, at temperaturen stiger hurtigere op til setpointet. Dog vil konsekvensen af dette give et overshoot, som tager en del tid om at ramme setpoint præcist. Mere om det senere.

Det var svært for os at sætte samplingsfrekvensen for måling af temperatursensoren over 3Hz, da databladet for LM75 siger, at der skal gå mindst 300 ms for hver aflæsning. Derfor har vi ikke forøget den.

Det kan dog naturligt udledes, at man vil få et mere retvisende billede af virkeligheden med flere samples per sekund. Herudover vil integralet over differensen mellem fejl og setpoint blive beregnet mere præcist, så der ikke er et så stort areal mellem hver sample, som ikke tælles med. Det kan påvirke temperaturreguleringen, så integrale leddet ikke dæmper sin vækst umiddelbart efter, at temperaturen nærmer sig setpoint. Herunder ses, hvordan integralet stiger markant efter hver sampling.

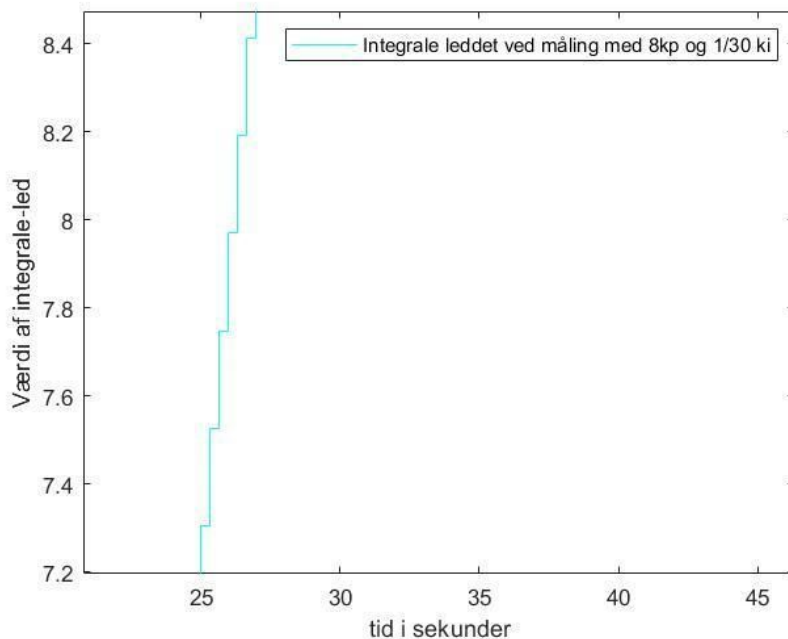


Figure 22 Zoom af graf for integrale-leddet ved den ene måling, hvor effekten af samplingsfrekvensen kommer til udtryk

Ulempen ved en høj samplingsfrekvens er, at man kan få kvantiseret støj, som egentlig bare skulle være udeladt. Dette temperaturreguleringssystem er et relativt langsomt system, så det lægger ikke op til tusindvis af ændringer for output-værdien i sekundet.

### Wind up

Integrale windup er blevet undgået ved at sænke værdien af  $K_i$  koefficienten. Det gør nemlig, at man undgår overshoot og ikke bruger unødvendig lang tid på, at temperaturen sænker sig ned til setpoint på 50 grader. Der ses her nogle eksempler med ens  $K_p$  koefficient og forskellig  $K_i$  koefficient:

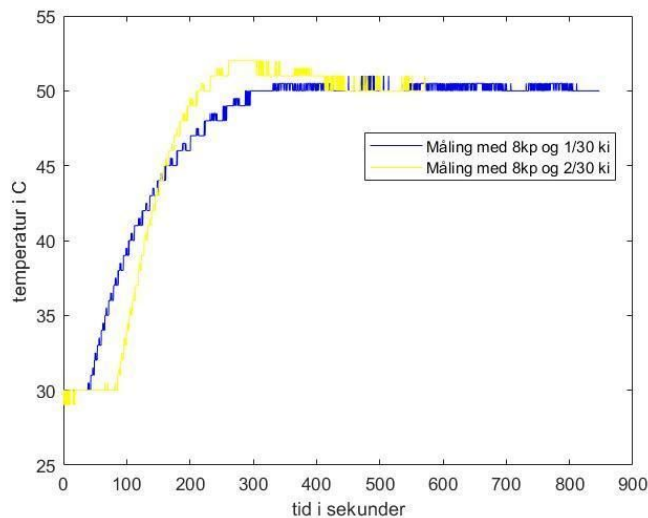


Figure 23 Graf for 2 målinger med forskellige Ki-værdier med startpunkt på 30 grader

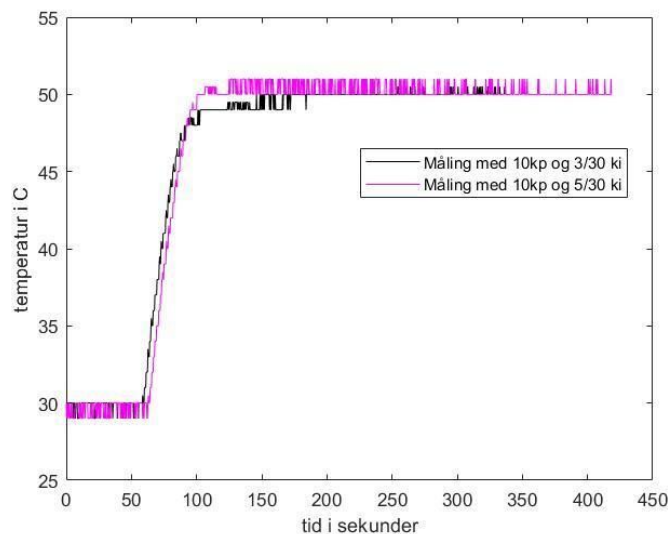


Figure 24 Graf for andre 2 målinger med forskellige Ki-værdier med startpunkt på 30 grader

Som det kan ses på figur 23, så stiger grafen med 2/30 ki hurtigere, men det giver et windup, som gør, at den er længere om at stabilisere sig på de 50 grader end den anden graf med ki på 1/30. Det er på trods af, at den gule graf er meget hurtigere til at ramme de 50 grader end den blå.

Der er ikke eksperimenteret med D-leddet i PID-systemet, så der kan ikke laves en konklusion på, hvad D skal være. Dog kan man generelt sige, at det differentierede led er lidt hurtigere at dæmpe ud ved setpoint i PD system end i et PI system. Dog har det nogle konsekvenser. Det er, at det kan få outputtet til at oscillere meget og give noget højfrekvent støj. Det kræver altså en høj samplingsfrekvens for, at det går godt, som vi ikke har i vores system. Dette temperaturreguleringsystem er også et relativt langsomt system, så det vil alligevel ikke lykkedes at få temperaturen til at svinge så hurtigt. Derved er effekten af det differentieret led ikke særlig markant, så en Kd på 0 er alligevel at foretrække for netop dette system.



# Konklusion

I denne øvelse er det lykkedes os at implementere en PID-controller til at fungere som en temperaturregulator. Ved at ændre på koefficienterne  $K_p$  og  $K_i$  er det lykkedes os at gå fra en temperatur på 30 grader celsius til 50 grader celsius. En stor  $K_p$ -værdi giver en stor vækst i starten efter setpoint er sat til 50 grader. Herefter vil  $K_i$ -leddet bidrage til en yderligere stigning, hvis den er høj, men til gengæld vil den give windup og bruge lang tid på at komme ned til setpoint. Derfor kunne en tilpas lav  $K_i$ -koefficient foretrækkes. Den skal være så høj, at den lige præcis ikke forårsager wind up. Vi fandt frem til, at en god kombination af  $K_p$  og  $K_i$  var en  $K_p$  på 10 og en  $K_i$  på 3/30.

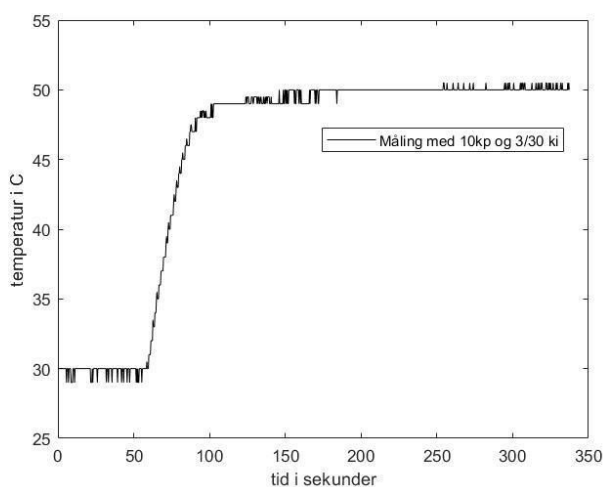


Figure 25 Graf for måling med de bedste  $K_i$ -koefficienter

Det kommer af, at den var hurtigst til at hæve temperaturen op på 50 grader og samtidig blive relativ stabil herefter.  $K_i$  var lige høj nok til, at wind up ikke forekom.