AARHUS UNIVERSITET

Aarhus University School of Engineering

GFV - Journal 3

Automatic Temperature Regulator

Mathias Martin Rahbek Markussen	Studienummer: 202107385	AU-id: au694672
Thomas Kaae	Studienummer: 202100350	AU-id: au698066
Shukriya Bile	Studienummer: 202107346	AU-id: au233984
Jakob Damgård Dall	Studienummer: 201805013	AU-id: au616329

DATO

March 31, 2022

Indholdsfortegnelse

1 Introduktion

Udarbejdelsen af denne journal og gennemførelsen af tilhørende laboratorieøvelse, har til sinde at skabe en grundlæggende forståelse for PID-controllere. Dette opnås gennem en række tests af en simpel controller.

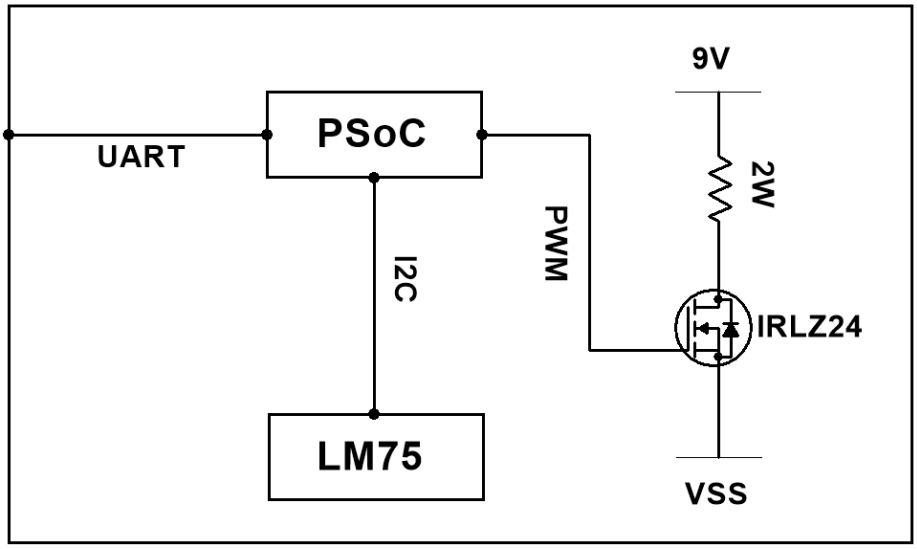
Til øvelsen gøres der brug af en LM75 temperatur sensor, en strømmodstand som aktuator, og den samme MOSFET, som der blev brugt i Journal 1.

Med I2C kommunikation aflæses den målte temperatur, hvor PSoC'en, med med en software PID-controller, regulerer PWM signalet, der sendes ud til aktuatoren, for at fastholde en given temperatur.

Gennem flere målinger varieres faktorernes scalering PID-controlleren, og resultatet heraf plottes. Ændringer i det proportionelle- og integrelle bidrag sammenlignes med det resulterende kontrolsignal ved forskellige værdier.
Slutteligt diskuteres sampling raten og dens betydning for kontrolsystemet, og det ønskes at finde frem til dén sammensætning af de forskellige bidrag, der giver det bedst mulige kontrolsystem.

2 Temperature Controller

2.1 Design



Figur 1: Kredsløb for øvelsen

2.1.1 Indledende overvejelser

At indstille en PID controller efter et bestemt system, er ikke en simpel opgave. Der er et hav a fremgangsmåder, af varierende kompleksitet.
I dette tilfælde er startværdier fundet ved brug af lampda tuning. Værdierne er opgivet af underviser.

En simpel PID-controller blev gjort tilgængelig af underviser. Denne kommer i form af et PSoC-creater projekt, med de essentielle funktioner til PID-controlleren inkluderet. Ydermere indeholdt projektet også funktionalitet til at printe relevante målinger på UART.
Her manglede dog I2C kommunikation med temperaturmåleren, samt UART kommunikation med computer. Det er altså disse to, er vil blive præsenteret i softwaredesign-afsnittet herunder.

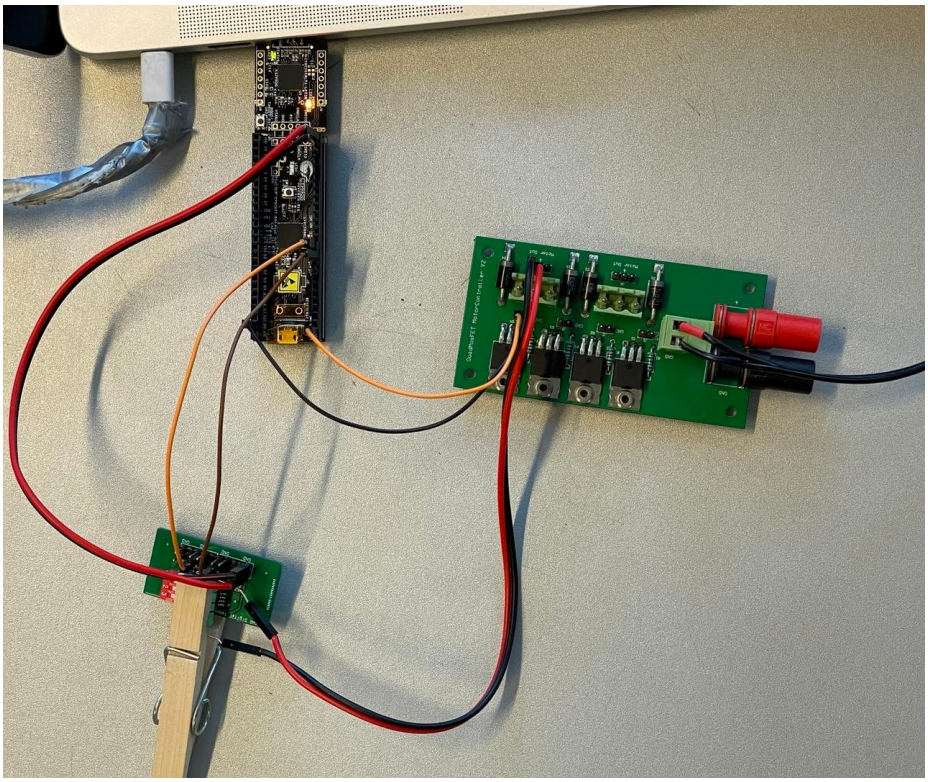
2.1.2 Softwaredesign

I2C kommunikation blev indført som i journal 2. Koden herfra blev genbrugt med kun få rettelser. Både 2's complement konvertering samt read funktioner genanvendes.
Koden til UART kommunikation er ligeledes genanvendt. Interrupt handler samt reaktion på input fra keyboard blev kopieret og redigeret, således at setpoint kunne ændres ved tryk på tastatur.
Målinger blev foretaget ved brug af RealTerm til pc.
Eftersom størstedelen af koden er genbrug, og præsenteret i forrige journaler, henvises til det vedhæftede projekt, hvis flere detaljer ønskes. Det vises ikke her.

2.1.3 Hardwaredesign

Navn	Port
Rx	P12[6]
Tx	P12[7]
SCL	P12[4]
SDA	P12[5]
controlSignal	P2[0]

Tabel 1: Forbindelser



Figur 2: Opstilling af øvelse

Til hardwaredesign anvendtes PSoC, LM75 temperatursensor, effektmodstand, samt MOSFET-printet.

Opstillingen ligner til forveksling den brugt i journal 1. Ændringen består i, at en effektmodstand er

placeret på temperatursensoren vha. en klemme . Effektmodstanden er koblet til MOSFET printet via gate'en. PSoC'en programmeres til at indeholde en UART TX og RX samt en I2C master. I2C masterens data- samt clock-forbindelse, tilknyttes fysiske pin på
PSoC'en henholdsvis P12[4] samt P12[5].

MOSFET-transistoren er som det visuelt fremgår af figur 1 forsynet via en ekstern kilde med en spændingsforsyning på 9V, som går til VSS. Via PSoC'en genereres der et PWM-signal til MOSFET-transistoren.

Med hjælp fra I2C-kommunikationen aflæses den målte temperatur. For at regulere eller fastholde en bestemt temperatur, varetages dette af PSoC'en, men med en software PID-controller, hvor PWM-signal der sendes til effektmodstanden reguleres. Effektmodstand fungerer som energikilde til at regulere temperaturen. Sammenhængen i dette er, at jo højere PWM-signal effektmodstande forsynes med, jo varmere bliver den. Da effektmodstanden er ovenpå temperatursensoren, er det givet, at jo varmere effektmodstanden er, desto højere temperatur vil temperatursensoren registrere.

Pinoversigten er vist på tabel 1.

2.2 Resultater

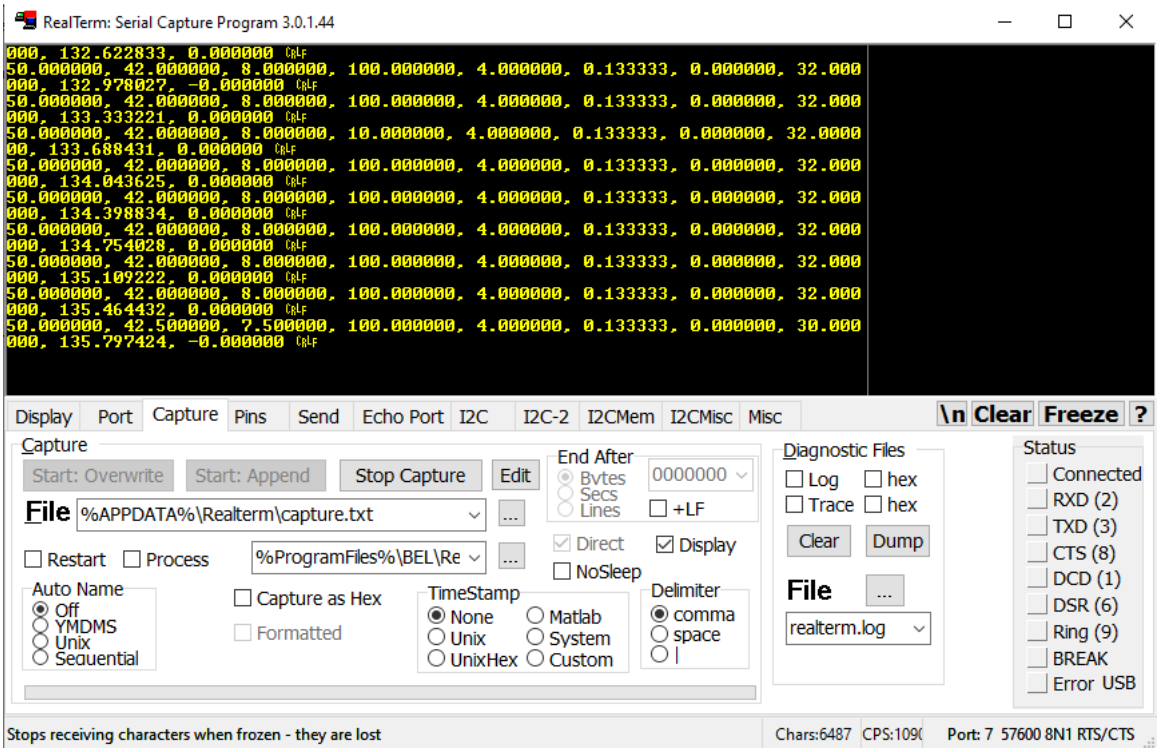
Målinger printes fra RealTerm i en .txt fil. Dataen er inddelt i søjler. RealTerm terminalen under måling er vist i figur 3, og udsnit fra en datafil er at se i figur 4.
Hver måling indlæses i MATLAB, hvor det behandles og forskønnes. Det blev valgt kun at plotte 10%, da temperatursensoren kun måler i trin af halve grader. Dette gav en meget "kantet" graf, der fjernede fokus fra det vigtige. Derudover er graferne plottet som en sammenhængende linje, fremfor individuelle målinger. Begge dele er gjort udelukkende for at forøge læsbarheden, og samtlige målinger er medtaget i eventuelle beregninger. Det skal ligeledes nævne, at tidsaksen i samtlige figurer, er udregnet fremfor målt. Her er det antaget, at der er præcist 3 eller 6 målinger pr sekund. Dette er dog ikke helt nøjagtigt, eftersom koden også tager en bestemt tid at køre - ud over de 300/600ms der ventes mellem målinger. Eftersom samtlige grafer behandles på samme måde, blev det vurderet, at den anvendte fremgangsmåde stadig var en tilstrækkelig måde at præsentere datapunkter.
Ydermere anvendes MATLAB til at aflæse rise time, overshoot og undershoot.
Rise time er her tiden fra ændring indtil setpoint temperaturen rammes første gang.
Overshoot er maksimale værdi over setpoint, der rammes.
Undershoot er mindste værdi der rammes, efter setpoint er ramt.
Se den samlede kode for MATLAB filen i appendix, code section 3.1. **Der laves i alt 8 målinger.**
Alle målinger foretages i ca. 20 minutter. Først 5 minutter til at ramme et setpoint på 30 grader, dernæst justeres setpoint til 50, og der måles i yderligere 15 minutter.

Først ændres proportionaldelens skalering systematisk.
Der foretages tre målinger, og mellem hver fordobles den. Se figur 5 til 10.

Dernæst ændres integraldelen.
Der foretages tre målinger, og mellem hver fordobles den. Se figur 5, og 6, samt 11 til 14.

Derefter plottes et gæt på optimale værdier, lavet ud fra de forrige målinger.
Se figurer 15 og 16.

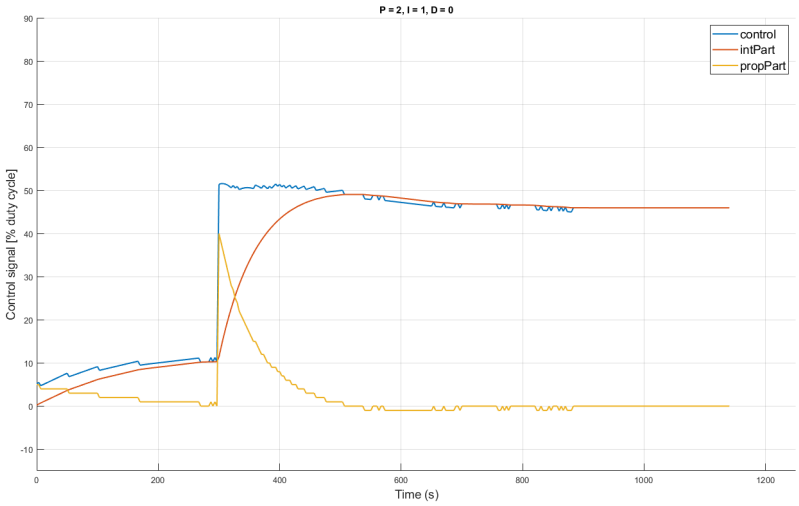
Slutteligt ændres sampling rate.
Først med en høj rate på 6/s, set på figur 17 og figur 18.
Dernæst en lav rate på 1/s, set på figur **??** og figur **??**.



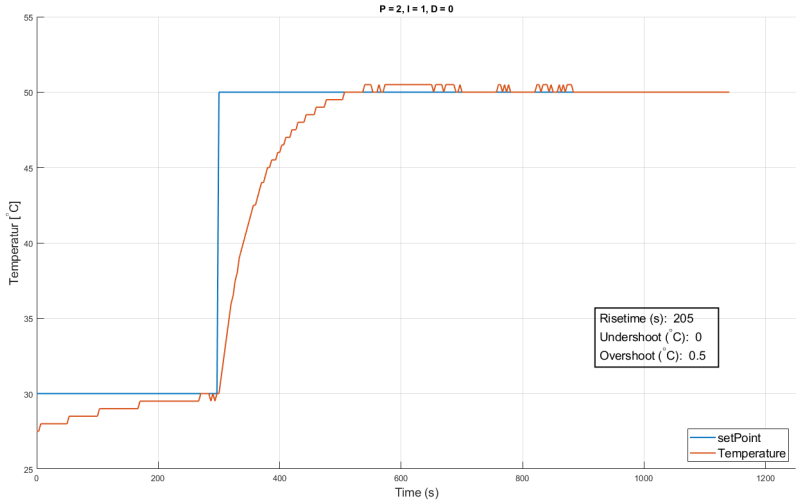
Figur 3: Terminal i RealTerm

setPoint	Current temp	Error	controlSignal	Kp	Ki	Kd	ProportinalPart	IntegralPart	DerivativePart
50.0000	35.0000	15.0000	84.2424	4.0000	0.1333	0.0000	60.0000	24.2424	0.0000
50.0000	35.0000	15.0000	84.9084	4.0000	0.1333	0.0000	60.0000	24.9084	0.0000
50.0000	35.5000	14.5000	83.5522	4.0000	0.1333	0.0000	58.0000	25.5522	0.0000
50.0000	35.5000	14.5000	84.1960	4.0000	0.1333	0.0000	58.0000	26.1960	0.0000
50.0000	35.5000	14.5000	84.8398	4.0000	0.1333	0.0000	58.0000	26.8398	0.0000
50.0000	36.0000	14.0000	83.4614	4.0000	0.1333	0.0000	56.0000	27.4614	0.0000
50.0000	36.0000	14.0000	84.0830	4.0000	0.1333	0.0000	56.0000	28.0830	0.0000
50.0000	36.5000	13.5000	82.6824	4.0000	0.1333	0.0000	54.0000	28.6824	0.0000
50.0000	36.5000	13.5000	83.2818	4.0000	0.1333	0.0000	54.0000	29.2818	0.0000
50.0000	37.0000	13.0000	81.8590	4.0000	0.1333	0.0000	52.0000	29.8590	0.0000
50.0000	37.0000	13.0000	82.4362	4.0000	0.1333	0.0000	52.0000	30.4362	0.0000
50.0000	37.0000	13.0000	83.0134	4.0000	0.1333	0.0000	52.0000	31.0134	0.0000
50.0000	37.5000	12.5000	81.5684	4.0000	0.1333	0.0000	50.0000	31.5684	0.0000
50.0000	37.5000	12.5000	82.1234	4.0000	0.1333	0.0000	50.0000	32.1234	0.0000
50.0000	38.0000	12.0000	80.6562	4.0000	0.1333	0.0000	48.0000	32.6562	0.0000
50.0000	38.0000	12.0000	81.1890	4.0000	0.1333	0.0000	48.0000	33.1890	0.0000
50.0000	38.0000	12.0000	81.7218	4.0000	0.1333	0.0000	48.0000	33.7218	0.0000
50.0000	38.5000	11.5000	80.2324	4.0000	0.1333	0.0000	46.0000	34.2324	0.0000
50.0000	38.5000	11.5000	80.7430	4.0000	0.1333	0.0000	46.0000	34.7430	0.0000
50.0000	38.5000	11.5000	81.2536	4.0000	0.1333	0.0000	46.0000	35.2536	0.0000
50.0000	39.0000	11.0000	79.7420	4.0000	0.1333	0.0000	44.0000	35.7420	0.0000
50.0000	39.0000	11.0000	80.2304	4.0000	0.1333	0.0000	44.0000	36.2304	0.0000
50.0000	39.0000	11.0000	80.7188	4.0000	0.1333	0.0000	44.0000	36.7188	0.0000
50.0000	39.5000	10.5000	79.1850	4.0000	0.1333	0.0000	42.0000	37.1850	0.0000
50.0000	39.5000	10.5000	79.6512	4.0000	0.1333	0.0000	42.0000	37.6512	0.0000

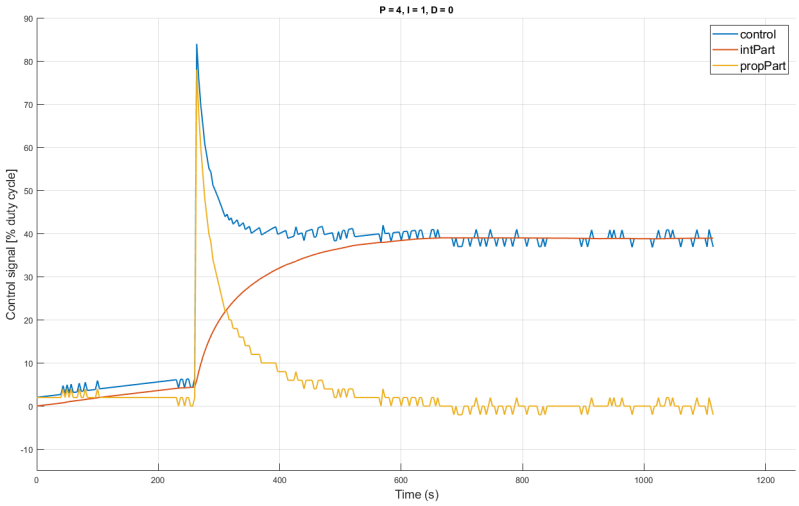
Figur 4: Udsnit af datafil



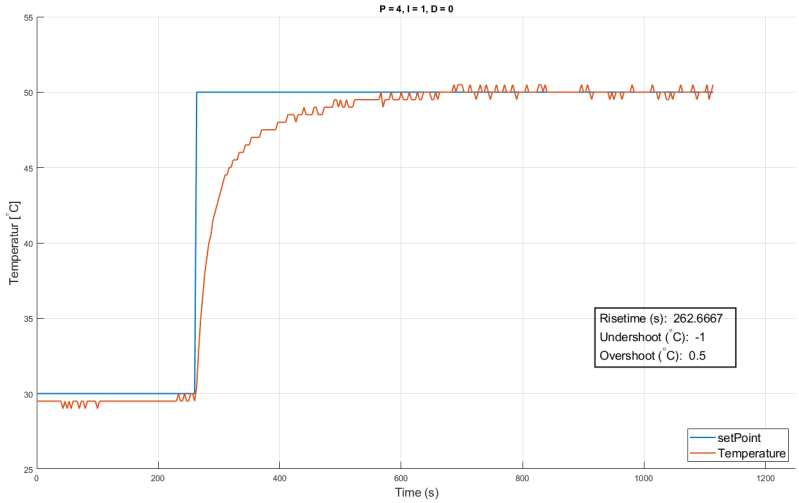
Figur 5: Måling 1: P = 2, I = 1, D = 0
Dokumentation af kontrol adfærd.
Indeledende tilstand kan ses i tidsintervallet [0 ;300]
Transient responsen kan ses i tidsintervallet [300 ; ∼600]
Steady-state responsen kan ses i intervallet [∼600 ; ∼1150]



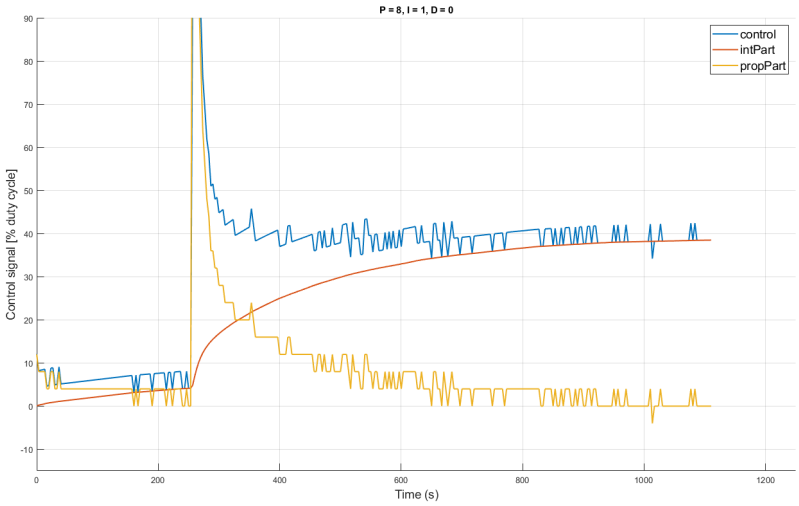
Figur 6: Måling 1: P = 2, I = 1, D = 0
Dokumentation af reference og målt temperatur.
Indeledende tilstand kan ses i tidsintervallet [0 ;300]
Transient responsen kan ses i tidsintervallet [300 ; ∼600]
Steady-state responsen kan ses i intervallet [∼600 ; ∼1150]



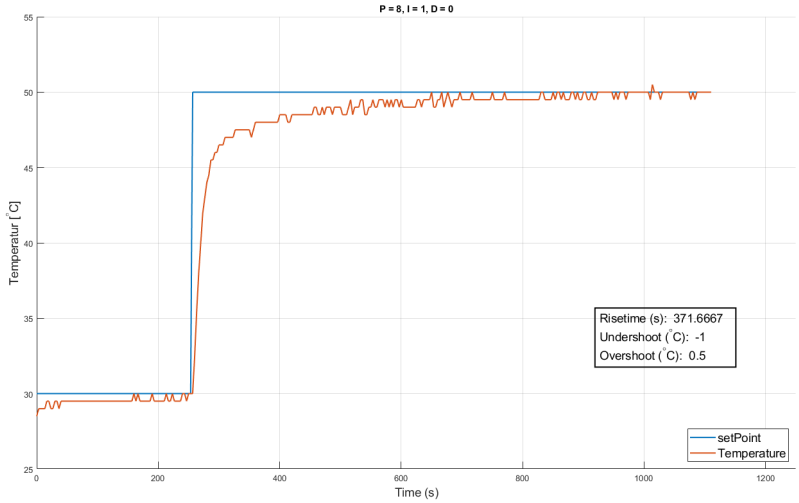
Figur 7: Måling 2: P = 4, I = 1, D = 0
Dokumentation af kontrol adfærd.
Indeledende tilstand kan ses i tidsintervallet [0; ∼250]
Transient responsen kan ses i tidsintervallet [∼250; ∼600]
Steady-state responsen kan ses i tidsintervallet [∼600; ∼1150]



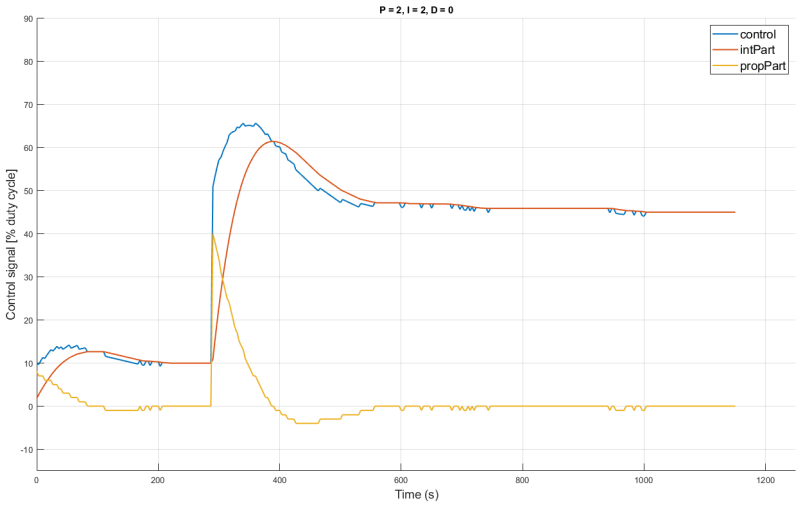
Figur 8: Måling 2: P = 4, I = 1, D = 0
Dokumentation af reference og målt temperatur.
Indeledende tilstand kan ses i tidsintervallet [0; ∼250]
Transient responsen kan ses i tidsintervallet [∼250; ∼600]
Steady-state responsen kan ses i intervallet [∼600; ∼1150]



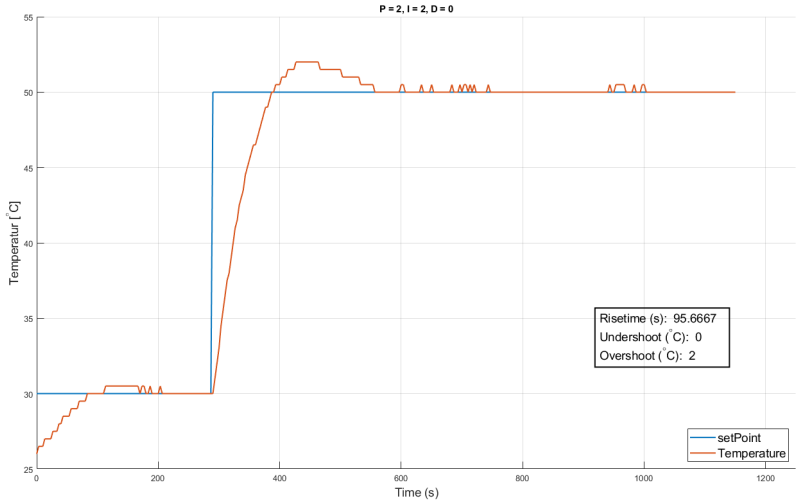
Figur 9: Måling 3: P = 8, I = 1, D = 0
Dokumentation af kontrol adfærd.
Indeledende tilstand kan ses i tidsintervallet [0; ∼250]
Transient responsen kan ses i tidsintervallet [∼250; ∼700]
Steady-state responsen kan ses i intervallet [∼700; ∼1150]



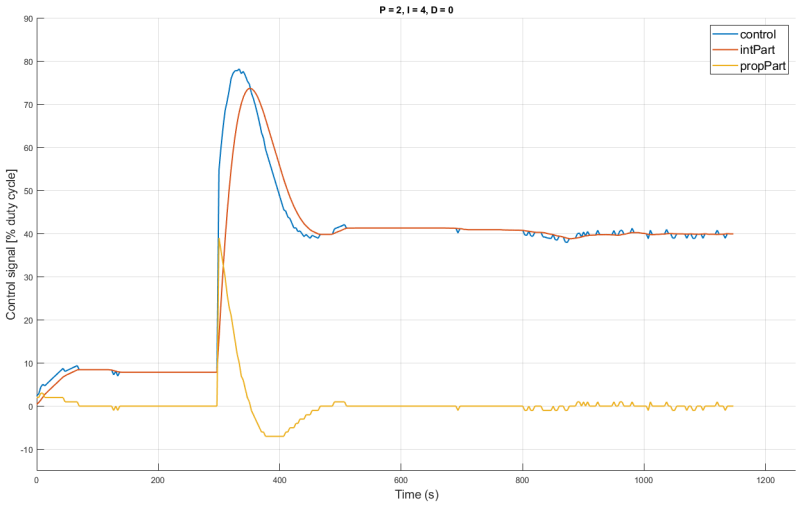
Figur 10: Måling 3: P = 8, I = 1, D = 0
Dokumentation af reference og målt temperatur.
Indeledende tilstand kan ses i tidsintervallet [0; ∼250]
Transient responsen kan ses i tidsintervallet [∼250; ∼700]
Steady-state responsen kan ses i intervallet [∼700; ∼1150]



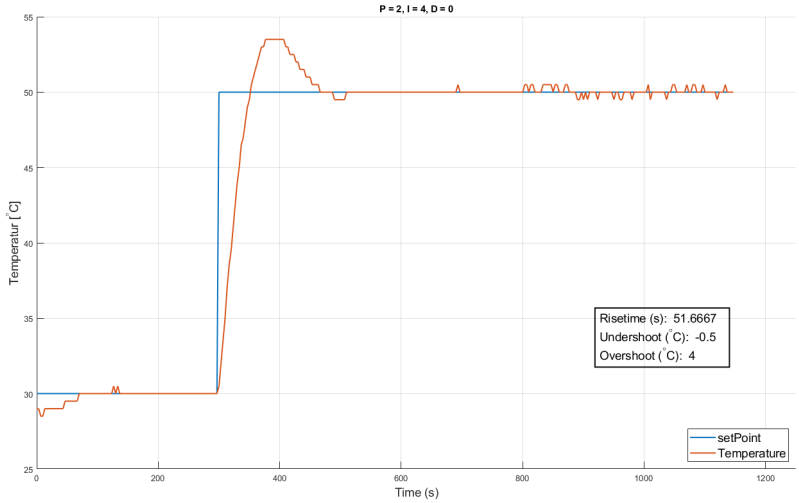
Figur 11: Måling 4: P = 2, I = 2, D = 0
Dokumentation af kontrol adfærd.
Indeledende tilstand kan ses i tidsintervallet [0; ∼300]
Transient responsen kan ses i tidsintervallet [∼300; ∼600]
Steady-state responsen kan ses i intervallet [∼600; ∼1175]



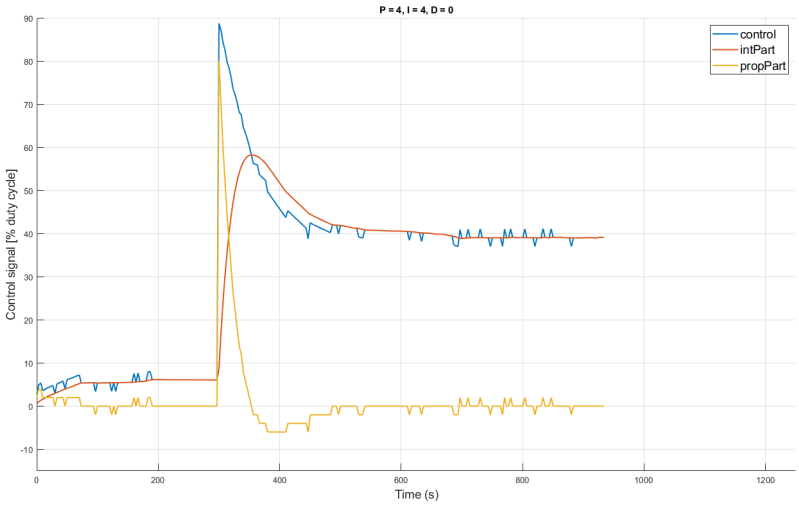
Figur 12: Måling 4: P = 2, I = 2, D = 0
Dokumentation af reference og målt temperatur.
Indeledende tilstand kan ses i tidsintervallet [0; ∼300]
Transient responsen kan ses i tidsintervallet [∼300; ∼600]
Steady-state responsen kan ses i intervallet [∼600; ∼1175]



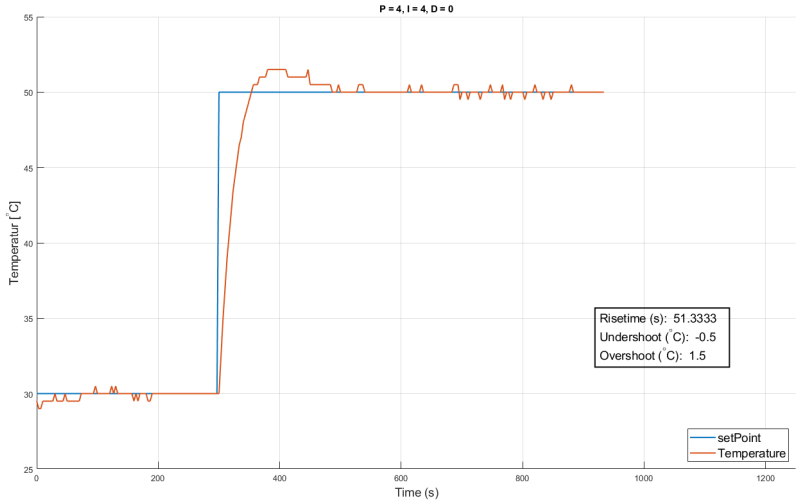
Figur 13: Måling 5: P = 2, I = 4, D = 0
Dokumentation af kontrol adfærd.
Indeledende tilstand kan ses i tidsintervallet [0 ; ∼300]
Transient responsen kan ses i tidsintervallet [∼300; ∼500]
Steady-state responsen kan ses i intervallet [∼500 ; ∼1175]



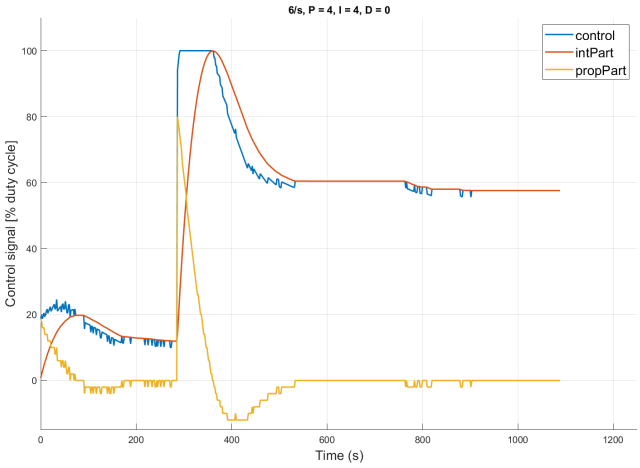
Figur 14: Måling 5: P = 2, I = 4, D = 0
Dokumentation af reference og målt temperatur.
Indeledende tilstand kan ses i tidsintervallet [0 ; ∼300]
Transient responsen kan ses i tidsintervallet [∼300; ∼500]
Steady-state responsen kan ses i intervallet [∼500 ; ∼1175]



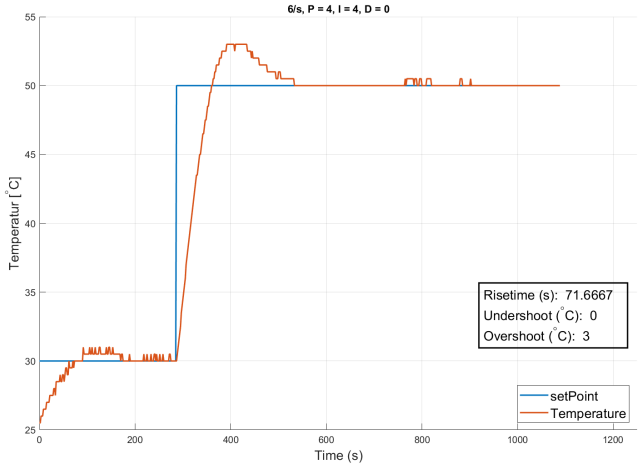
Figur 15: Måling 6: P = 4, I = 4, D = 0
Dokumentation af kontrol adfærd.
Indeledende tilstand kan ses i tidsintervallet [0 ; ∼300]
Transient responsen kan ses i tidsintervallet [∼300; ∼500]
Steady-state responsen kan ses i intervallet [∼500 ; ∼925]



Figur 16: Måling 6: P = 4, I = 4, D = 0
Dokumentation af reference og målt temperatur.
Indeledende tilstand kan ses i tidsintervallet [0 ; ∼300]
Transient responsen kan ses i tidsintervallet [∼300; ∼500]
Steady-state responsen kan ses i intervallet [∼500 ; ∼925]



Figur 17: Måling 7: SampleRate = 6/s, P = 4, I = 4, D = 0
Dokumentation af kontrol adfærd.
Indeledende tilstand kan ses i tidsintervallet [0 ; ∼290]
Transient responsen kan ses i tidsintervallet [290; ∼500]
Steady-state responsen kan ses i intervallet [∼500; ∼1225]



Figur 18: Måling 7: SampleRate = 6/s, P = 4, I = 4, D = 0
Dokumentation af reference og målt temperatur.
Indeledende tilstand kan ses i tidsintervallet [0 ; ∼290]
Transient responsen kan ses i tidsintervallet [290; ∼500]
Steady-state responsen kan ses i intervallet [500; ∼1225]

2.3 Diskussion

Kortfattet er følgende observeret ud fra seks målinger:

Måling Nr.	Rise Time	Overshoot	Settling Time
1	205s	0.5°C	↑
2	262.7s	0.5°C	↑↑
3	371.7s	0.5°C	↑↑↑
4	95.7s	2°C	↑↑
5	51.7s	4°C	↑↑↑
6	51.3s	1.5°C	↑
7	71.7s	3°C	↑↑↑

Tabel 2: Skalering af proportionaldel og integraledel og indvirkningen på risetime, overshot og settling time

Målingerne viser både, at PID controlleren er i stand til at regulere temperaturen adaptivt. Det vises også hvordan den er i stand til dette. Ved beskrivelsen af controlsignalet, kan dets bidrag analyseres hver for sig, og effekten af faktorenes skalering analyseres.

Det ses, at risetime vokser når Kp vokser. Dette går umiddelbart imod intuition. Det skyldes den måde risetime udregnes på. Den regnes som tiden fra setpoint ændres til den nye temperatur rammes. Proportionaldelen er effektiv når error er høj. Derfor vokser temperaturen først meget hurtigt, men tager derefter lang tid om at ramme den ønskede. Var risetime defineret som tiden til 0.8*setpoint, ville det være mere repræsentativt for Kp faktorens indflydelse. Dette kan gøres i en fremtidig undersøgelse.

Som Ki gøres stor, sker en del ændringer. Rise time falder, og overshoot og undershoot stiger markant. Integralbidraget til controlsignalet vokser hurtigt, når error er høj. Selvom error falder, 'husker' integralet stadig, at det var højt før. Dette fører til, at der skydes en del over den ønskede temperatur. Her bliver integralet negativt, og falder igen. Når integralet at falde for meget, skydes der under, og det samme fortsættes.

Et gæt på en optimeret PID controller er et mix af de to værdier. Det ses, at den har en lav risetime, og ikke skyder for langt over. Dette er dog bare et umiddelbart gæt, og kan let optimeres yderligere. Det vurderes dog som værende uden for rammerne af denne øvelse. Mange flere målinger ville være nødvendige, og der vil også skulle eksperimenteres med differentialebidraget.

De sidste målinger relaterer sig til samplerate. Her blev foretaget en måling, hvor den tog 6 målinger

pr sekund. Først og fremmest skal det nævnes, at databladet for temperatursensoren, kun garanterer korrekte målinger 3 gange i sekundet. Der kan derfor forventes samme målinger i streg, eller anden udokumenteret opførsel.
Og eftersom integraldelen udregnes for hver målepunkt, vil en forøgning i sample rate medføre, at den vokser markant hurtigere. Temperaturen vil ikke nødvendigvis blive reguleret hurtigere med en lige så stor faktor, og resultatet bliver at integraldelen kommer til at være dominerende for kontrolsignalet. De typiske tegn på en for stor integraldel vil således forekomme: kontrolsignalet svinger omkring setpoint, et stort overshoot, og er længere tid om at stabiliseres. Dette ses også på figur 17, hvor integraledelen svinger en del i starten af målingerne.

2.4 Konklusion

Der er lykkedes at bruge PID-controlleren til regulering af temperaturen aflæst af LM75.

Det ses i målingerne, at der ved ændring i faktorenes scalering i PID-controlleren, sker en tydelig ændring i det resulterende kontrolsignal og temperatur-stigningen.

Med to målinger er sampling ratens betydning for kontrolsystemet blevet undersøgt, og resultatet af dette er blevet diskuteret.

Der er ud fra målingerne blevet givet et bud på dén sammensætning af de forskellige bidrag, der resulterer i et optimalt kontrolsystem..

3 Appendix

MATLAB kode til plots

```
clear all; close all; clc;
set(0,'DefaultFigureWindowStyle','docked');
set(0, 'DefaultLineLineWidth', 1.5);

% MÅLINGER MED SAMPLE RATE PÅ CA. 3 PR SEKUND. HUSK AT ÆNDRE SAMPLE
% VARIABEL, HVIS ANDEN ANVENDES
plotandprint('30til50_P2_I1_d0.txt','P = 2, I = 1, D = 0')

plotandprint('30til50_P4_I1_d0.txt','P = 4, I = 1, D = 0')

plotandprint('30til50_P8_I1_d0.txt','P = 8, I = 1, D = 0')

plotandprint('30til50_P2_I2_d0.txt','P = 2, I = 2, D = 0')

plotandprint('30til50_P2_I4_D0.txt','P = 2, I = 4, D = 0')

plotandprint('30til50_P4_I4_D0.txt','P = 4, I = 4, D = 0')

```

```
19
20  % MÅLINGER MED SAMPLE RATE PÅ CA. 3 PR SEKUND. HUSK AT ÆNDRE SAMPLE
21  % VARIABEL, HVIS ANDEN ANVENDES
22  %plotandprint
23  %('30til50_P4_I4_D0_Integrale1000_Måling6.txt','6/s, P = 4, I = 4, D = 0')
24
25
26
27  function plotandprint(filestr,titlestr)
28  points = 10;
29  samplesprsek = 3;
30
31
32  A = importdata(filestr);
33  setPoint = A(:,1);
34  temp = A(:,2);
35  error = A(:,3);
36  control = A(:,4);
37  propPart = A(:,8);
38  intPart = A(:,9);
39  derivPart = A(:,10);
40  time = linspace(0,1/samplesprsek*length(temp),length(temp));
41
42  setPointplot = setPoint(1:points:(length(temp)))
43  tempplot = temp(1:points:length(temp));
44  errorplot = error(1:points:length(temp));
45  controlplot = control(1:points:length(temp));
46  propPartplot = propPart(1:points:length(temp));
47  intPartplot = intPart(1:points:length(temp));
48  derivPartplot = derivPart(1:points:length(temp));
49  timeplot = time(1:points:length(temp));
50
51
52  disp('Approximate rise time in seconds')
53  last30 = find(temp == 30);
54  first50 = find(temp == 50);
55  risetime = (first50(1) - last30(end))/samplesprsek;
56
57  disp('Maximum overshoot recorded')
58  overshoot = max(temp) - 50;
59  disp('Maximum undershoot recorded')
60  undershoot = min(temp(first50:end)) - 50;
61
62
63  figure
64  hold on
65  title(titlestr);
```

```
66  grid on;
67  xlabel('Time (s)', 'FontSize', 14);
68  ylabel('Control signal [% duty cycle]', 'FontSize', 14);
69  plot(timeplot,controlplot);
70  plot(timeplot,intPartplot);
71  plot(timeplot,propPartplot);
72  axis([0 1250 -15 110])
73  [~,b] = legend( 'control', 'intPart', 'propPart', 'Location',...
74      'NorthEast', 'Interpreter', 'none' , 'FontSize', 14);
75  set(findobj(b,'-property','MarkerSize'),'MarkerSize',30)
76
77  figure
78  hold on
79  title(titlestr);
80  grid on;
81  xlabel('Time (s)', 'FontSize', 14);
82  ylabel('Temperatur [^\circC]', 'FontSize', 14);
83  plot(timeplot,setPointplot);
84  plot(timeplot,tempplot);
85  axis([0 1250 25 55])
86  [~,b] = legend('setPoint', 'Temperature', 'Location', 'SouthEast',...
87      'Interpreter', 'none', 'FontSize', 14);
88  set(findobj(b,'-property','MarkerSize'),'MarkerSize',30)
89
90  annotation('textbox', [0.7, 0.3, 0.1, 0.1], 'String', ...
91      ['Risetime (s):  ', num2str(risetime), ...
92      newline, 'Undershoot (^\circC):  ', num2str(undershoot), newline, ...
93      'Overshoot (^\circC):  ', num2str(overshoot)], 'FontSize', 14);
94
95  end
```