



DET TEKNISK-NATURVITENSKAPELIGE FAKULTET

BACHELOROPPGAVE

Studieprogram/spesialisering:	Vårsemesteret 2022
Bachelor i ingeniørfag / Automatisering og elektronikkdesign	Åpen
Forfattere: Sondre Salvesen, Sindre Mikkelsen & André Risa Rabba	
Fagansvarlig: Tormod Drengstig	
Veileder: Tormod Drengstig	
Tittel på bacheloroppgaven: Bruk av Sysmac Studio sammen med Simulink	
Engelsk tittel: The use of Sysmac Studio together with Simulink	
Studiepoeng: 20	
Emneord: Simulink, PLC Coder, PLS, HMI, MATLAB, Regulering, Sysmac Studio	Sidetall: 49 + vedlegg/annet: 40 Stavanger 15. mai 2022

Innhold

Innhold	i
Sammendrag	vii
1 Introduksjon	1
1.1 Problemstilling	1
1.2 Bacheloroppgavens struktur	2
2 Systembeskrivelse og ombygging	3
2.1 Tretankanlegget	3
2.2 Design av HMI	6
2.3 PLS Programmering	6
2.3.1 IO-Link	7
3 Modell og simulering	8
3.1 Pumpekaraktistikk	8

INNHOOLD

3.2	Ventilkarakteristikk	12
3.3	Modellering	16
3.4	Simulinkmodell	18
3.5	Verifisering	20
4	PLC Coder og Sysmac Studio	24
4.1	PLC Coder	24
4.1.1	Generering av kode til Sysmac Studio	25
4.1.2	Importere strukturert tekst fra Simulink inn i Sysmac Studio	32
4.2	Justeringer av generert kode	34
4.2.1	Variabler	34
4.2.2	Endring av kode i funksjonsblokken	35
5	Simulering og regulering i Simulink	36
5.1	Nivåregulering tank 1: Pumpe P01	36
5.2	Nivåregulering tank 1: Ventil LV11	38
6	Regulering mot virkelig prosess	44
6.1	Reguleringsproblematikk med overført pumperegulator	44
6.2	Overført regulator: pumpe P01	47
6.3	Overført regulator: ventil LV11	48

INNHold

7 Konklusjon	49
Bibliografi	50
Vedlegg	50
A Programlisting	51
A.1 Overleaf	51
A.2 Matlab	51
A.3 Simulink	51
A.3.1 Simulink PLC Coder	51
A.4 Sysmac studio	51
B Ombygging	52
B.1 Ombygging av styreskap	52
B.1.1 Inmat	52
B.1.2 Skapdør	55
B.2 Tretank oppgradering	59
B.2.1 Utbedring av eksisterende tretank anlegg	59
B.2.2 Montering av endebrytere til LV12, LV22 og LV32	59
B.2.3 Omkobling av ventiler	61
B.3 Feilsøking	62

INNHold

B.3.1	Feilsøking på pumpene	62
B.3.2	Feilsøking på DP Celle	67
C	Datablad	71
C.1	Pumper	71
C.1.1	Datablad til pumpe P01 og P02	71
C.1.2	Datablad til defekte pumper	71
C.2	Ventiler	71
C.2.1	Datablad til ventil LV11, LV21 og LV31	71
C.3	Nivåmålere	71
C.3.1	Datablad I/O-link nivåmåler	71
C.3.2	Datablad til nivåmåler LT1, LT2 og LT3	71
C.4	Effektregulator	72
C.4.1	Datablad PWM Effektregulator	72
C.5	Omron komponenter	72
C.5.1	Datablad HMI	72
C.5.2	Datablad CPU	72
C.5.3	Datablad AI/AO Moduler	72
C.5.4	Datablad DI/DO Moduler	72
C.5.5	Datablad IO-link master modul	72

INNHold

C.5.6	Datablad I/O Power Modul	72
C.5.7	Datablad Switch	72
C.5.8	Datablad IO-Link System	72
C.5.9	Datablad 24V powersupply	72
C.5.10	Omron CJ1	72
C.6	Diverse Komponenter	73
C.6.1	Datablad Sukkerbit	73
C.6.2	Datablad ABB AP-9	73
C.6.3	Datablad Rekkeklemme	73
C.6.4	Datablad Stripsfeste	73
C.6.5	Datablad Endebryter	73
C.6.6	Datablad Lampe	73
D	Skjematiske tegninger	74
E	Brukermanual	75
F	PLS program	76
F.1	Beskrivelse	76
F.2	Sysmac Studio kode	76
G	Matlab filer	77

INNHold

H Kode	78
H.1 Kode for tretankanlegg verdier	78
H.2 Kode i Matlabfunction	81
H.3 Kode for verifisering	82
H.3.1 Verifisering tank 1	82
H.3.2 Verifisering tank 2	83
H.3.3 Verifisering tank 3	84
H.4 Generert kode for pumpekontroller	84
H.5 Generert kode for ventilkontroller	86
H.6 Kode for plott av resultat	88
H.6.1 Kode for pumperegulator plott	88
H.6.2 Kode for ventilregulator plott	89

Sammendrag

I denne bacheloroppgaven undersøkte vi om Sysmac Studio kan brukes sammen med Simulink. I Simulink brukte vi tilleggspakken Simulink PLC Coder som gjør det mulig å konvertere simulinkmodeller til strukturert tekst(ST). For å teste den genererte ST koden anvendte vi et fysisk tretankanlegg som sto på et laboratorie ved Instituttet for Data- og Elektro. Det fysiske tretankanlegget ble oppgradert med nye komponenter, anlegget kan nå brukes i undervisning. Det ble utviklet et nytt PLS og HMI program som gjør det mulig å styre og regulere prosessen til tretankanlegget. Vi konstruerte en matematisk modell av tretankanlegget som gjør det mulig å simulere prosessen. Reguleringsmetodene som ble undersøkt i denne bacheloroppgaven var enkel nivåregulering for en tank ved hjelp av en pumperegulator(PI integrator) og en mer avanseert ventilregulator(**Gain Scheduling**). Vi genererte ST kode for pumperegulatoren og sammenliknet den med en innebygd regulator i Sysmac Studio for å teste Simulink PLC Coder. Vi valgte en mer avansert regulator av ventilen for å vise nytteverdien til Simulink PLC Coder ved mer avanserte beregninger. Vi konkluderte med at verktøyet Simulink PLC Coder fungerer, men at det er mer tidkrevende å bruke for enkle beregninger. Ved utføring av avanserte beregninger, som for eksempel **Gain Scheduling** er Simulink PLC Coder et mer nyttig verktøy.

Kapittel 1

Introduksjon

1.1 Problemstilling

Institutt for data- og elektroteknologi har oppgradert PLS'ene sine fra CJ1 (se vedlegg C.5.10) til NX102-1000 (se vedlegg C.5.2). Instituttet har også en prosessmodell bestående av tre tanker som styres/reguleres av en CJ1-PLS. Oppgaven går ut på å bygge om styreskapet til denne tretankprosessen for å bruke den nye NX102-1000. Siden det er en stund siden tretankprosessen var i drift, må det tekniske utstyret sjekkes og eventuelt oppgraderes. Vi skal se på muligheter for å bruke Matlab programmet Simulink sammen med Omron programmet Sysmac studio ved hjelp av verktøyet Simulink PLC Coder. Det skal konkret utvikles kode, fra Simulink PLC Coder, for nivåregulering av en tretankprosess. Parameterne til PI-regulatorene bestemmes ut fra en matematisk modell av tretanken implementert i Simulink. Ved å benytte den matematiske modellen skal det i Simulink simuleres forventede responser ved sprang i nivåreferansene. Responsene skal deretter verifiseres på det fysiske anlegget. Oppgaven går også ut på å utvikle nytt program og grensesnitt til HMI skjermen i styreskapet slik at regulatorparametre kan spesifiseres fra skjermen.

1.2 Bacheloroppgavens struktur

1.2 Bacheloroppgavens struktur

Strukturen til bacheloroppgaven er som følger:

2. Gjennomgang av systemets oppbygging, oppgraderinger og endringer av system og programvare.
3. Konstruksjon av matematisk modell og simuleringer.
4. Virkemåten til Simulink PLC Coder og hvordan det kan brukes sammen med Sysmac Studio.
5. Teoretisk regulering av tretankprosessen.
6. Regulering mot virkelig prosess og resultatene av å overføre en generert kode.
7. Refleksjon av arbeidet og resultat.

Kapittel 2

Systembeskrivelse og ombygging

I dette kapittelet kommer det en overordnet beskrivelse av tretankanlegget og hvilke oppgraderinger som er gjort. I tillegg kommer det en oversikt over HMI-skjerm og PLS-kode.

2.1 Tretankanlegget

Figur 2.1 representerer tretankanleggets oppbygning og funksjonalitet. Pumpe P01 forsyner tank 1 og pumpe P02 forsyner tank 2. Det er manuelle ventiler (LV12, LV22 & LV32) ut fra hver enkelt tank, ventilene representerer forstyrrelse på systemet. I hver tank er det fastmontert nivåålere (LT1, LT2 & LT3). Fra tank 1 til tank 2, tank 2 til tank 3 og ut ifra tank 3 er det montert ventiler (LV11, LV21 & LV31) som kan åpnes manuelt eller automatisk. Når væsken tappes ut av hver enkelt tank ender det opp i oppsamlingskaret i bunnen, hvor pumpe P01 og P02 sirkulerer det tilbake til enten tank 1 eller tank 2 når pumpene blir aktivert.

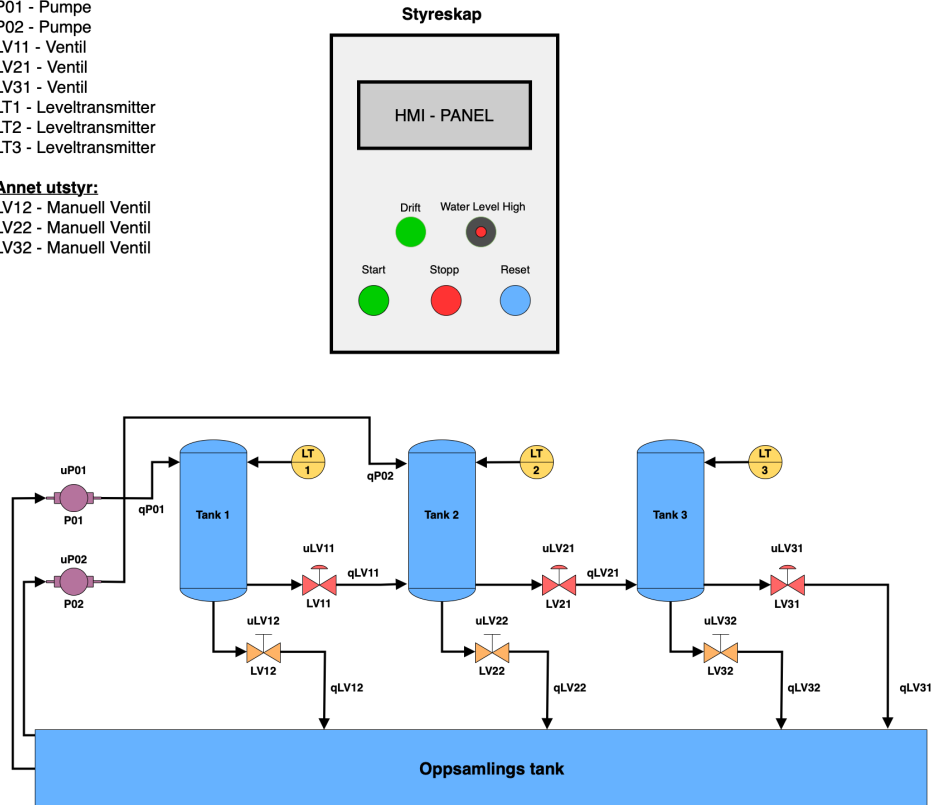
2.1 Tretankanlegget

Komponenter som blir styrt fra styreskap:

P01 - Pumpe
P02 - Pumpe
LV11 - Ventil
LV21 - Ventil
LV31 - Ventil
LT1 - Leveltransmitter
LT2 - Leveltransmitter
LT3 - Leveltransmitter

Annet utstyr:

LV12 - Manuell Ventil
LV22 - Manuell Ventil
LV32 - Manuell Ventil



Figur 2.1: Overordnet skisse av tretank systemet.

For å bruke tretankanlegget utførte vi flere oppgraderinger, dette har vært en svært omfattende og tidkrevende del av bacheloroppgaven. Punktlisten under viser oppgraderingene som ble utført, for en mer omfattende og detaljert beskrivelse se vedlegg B.

2.1 Tretankanlegget

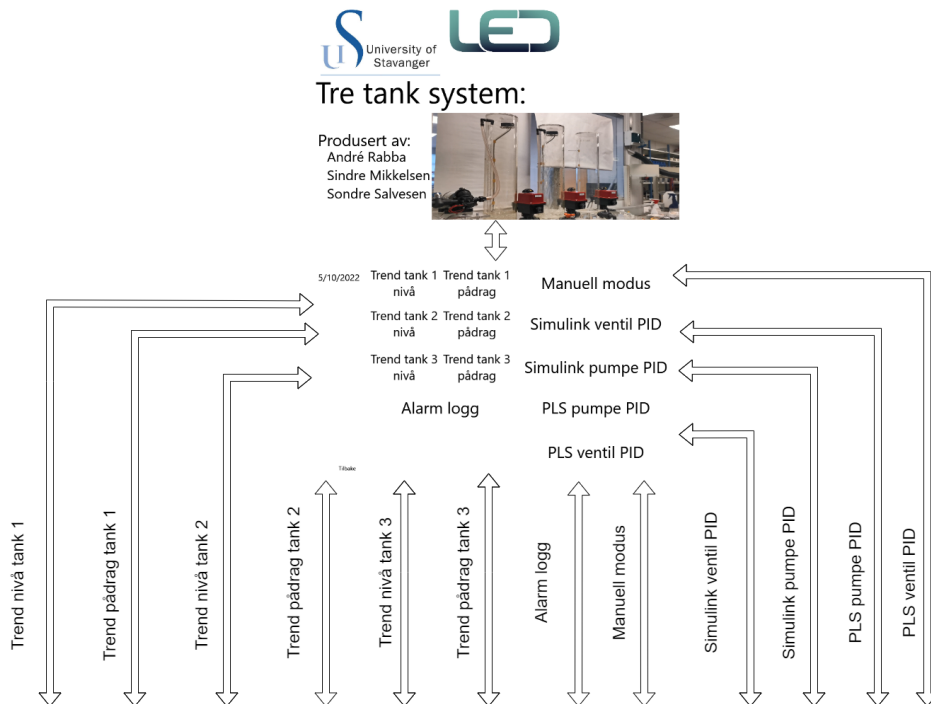
Liste over oppgraderinger som ble utført:

- Ny HMI skjerm (NA5-7W001S-V1). Se vedlegg C.5.1
- Nytt HMI brukergrensesnitt. Se vedlegg E
- Ny PLS (NX102-1000). Se vedlegg C.5.2
- Nytt PLS program. Se vedlegg F.2
- Nye elektriske PCSHEMATIC tegninger. Se vedlegg D
- Nye Analoge inngangs og -utgangsmoduler (AD3603, DA3603). Se vedlegg C.5.3
- Nye Digitale inngangs og -utgangsmoduler (ID5442, OD5256). Se vedlegg C.5.4
- Power supply (OMRON S8VS-06024B). Se vedlegg C.5.6
- Switch (OMRON W4S1-05B). Se vedlegg C.5.7
- Nye PWM regulatorer (DC Motor Speed Regulator 6V-90V 15A). Se vedlegg C.4.1
- Nye rekkeklemmer. Se vedlegg C.6.3
- Nye pumper (Vaakumpumpe Biltema Art. 25-999). Se vedlegg C.1.1
- Nye nivåmålere (Turck PT0.25R-1004-IOL-H1141). Se vedlegg C.3.1
- Konfigurasjon av I/O - link. Se vedlegg C.5.8
- Ny lampe som indikerer drift (Schneider XB4BVB3). Se vedlegg C.6.6
- Omkobling av automatiske ventiler LV11, LV21 og LV31 (VALPES ER20.X3B.GP6). Se vedlegg C.2.1
- Montering av endebrytere til manuelle ventiler LV12, LV22 og LV32 (Panasonic AV34243). Se vedlegg C.6.5
- 3D printet festeanordning til endebrytere.
- Fjerning av vribryter i skapdør.
- Oppgradering av eksisterende koblinger på tretank anlegget. Se vedlegg D
- Stripsfester. Se vedlegg C.6.4
- Koblingsbokser (ABB AP-9 IP65). Se vedlegg C.6.2
- Nye sukkerbiter (ENSTO KA460.12). Se vedlegg C.6.1

2.2 Design av HMI

2.2 Design av HMI

Vi har designet et grafisk brukergrensesnitt for gjøre det mulig å betjene og lese av data fra tretankanlegget. Se brukermanual for mer informasjon om virkemåten til de ulike skjermbildene i vedlegg E.



Figur 2.2: Overordnet skisse av brukergrensesnittet.

2.3 PLS Programmering

Vi programmerte et nytt PLS program for å kunne regulere tretank anlegget med PLS og Simulink PLC Coder. For mer detaljer om virkemåten se PLS programmet i sin helhet i vedlegg F.2. PLS programmet er også utviklet slik at det er mulig å laste inn nye funksjonsblokker som blir laget i Simulink. De kan deretter benyttes direkte uten at det er et behov for å gjøre endringer i PLS programmet.

2.3 PLS Programmering

2.3.1 IO-Link

Det var en del utfordringer med kommunikasjonen mellom IO-link mastere og IO-link komponenten. For å utbedre disse feilene kommuniserte vi med Omron og leverandøren av IO-link komponenten. For mer informasjon om hvordan utstyret ble konfigurert og kommunikasjonsproblematikken ble utbedret se vedlegg C.5.8 og C.5.5.

Kapittel 3

Modell og simulering

For å modellere og simulere systemet ble det beregnet pumpe og ventilkarakteristik. Ved hjelp av karakteristikken ble den matematiske modellen konstruert. Den matematiske modellen ble brukt til å få innsikt i dynamikken til prosessen, verifisere modellen og bestemme regulatorparametrene.

3.1 Pumpekarakteristikk

For å simulere pumpene i Simulink beregnet vi pumpekarakteristikk. Pumpekarakteristikken ble funnet ved å måle tiden det tar å fylle en av tankene ved forskjellig pumpepådrag som ble brukt til å beregne volumstrømmen til pumpene. På grunn av høydeforskjell mellom pumpene og vanninntaket til tankene vist i figur 3.1, var det nødvendig med et pumpepådrag på 17% for å få innstrømning til tankene.

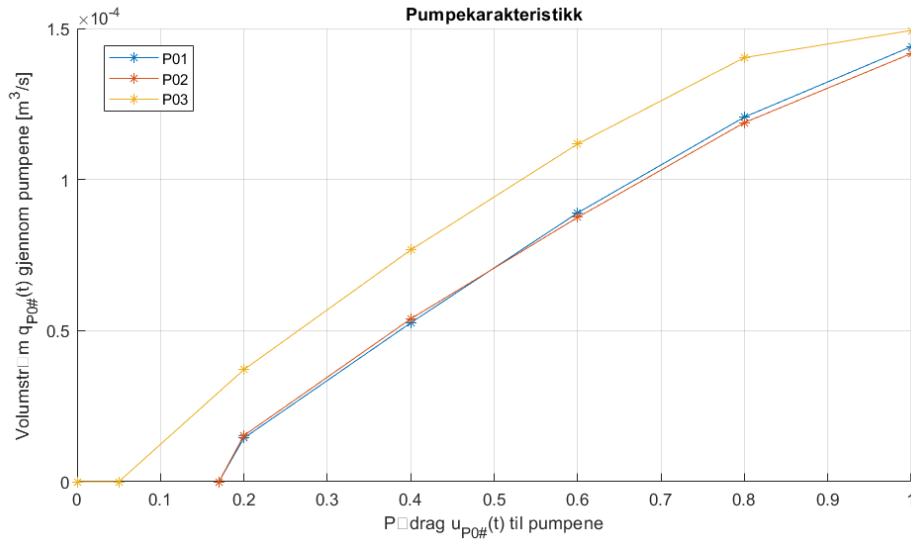
3.1 Pumpekarakteristikk



Figur 3.1: Høydedifferanse mellom pumpe og tank.

Siden tank 3 ikke får tilført vann fra en pumpe på samme måte som tank 1 og tank 2, ble det lagt et rør fra pumpe P02 til tank 3. Vi beregnet pumpekarakteristikken til det nye oppsettet til tank 3. Pumpekarakteristikken til pumpene blir brukt til å beregne ventilkarakteristikken i seksjon 3.2. I figur 3.2 kan man se de forskjellige pumpekarakteristikkene.

3.1 Pumpekarakteristikk



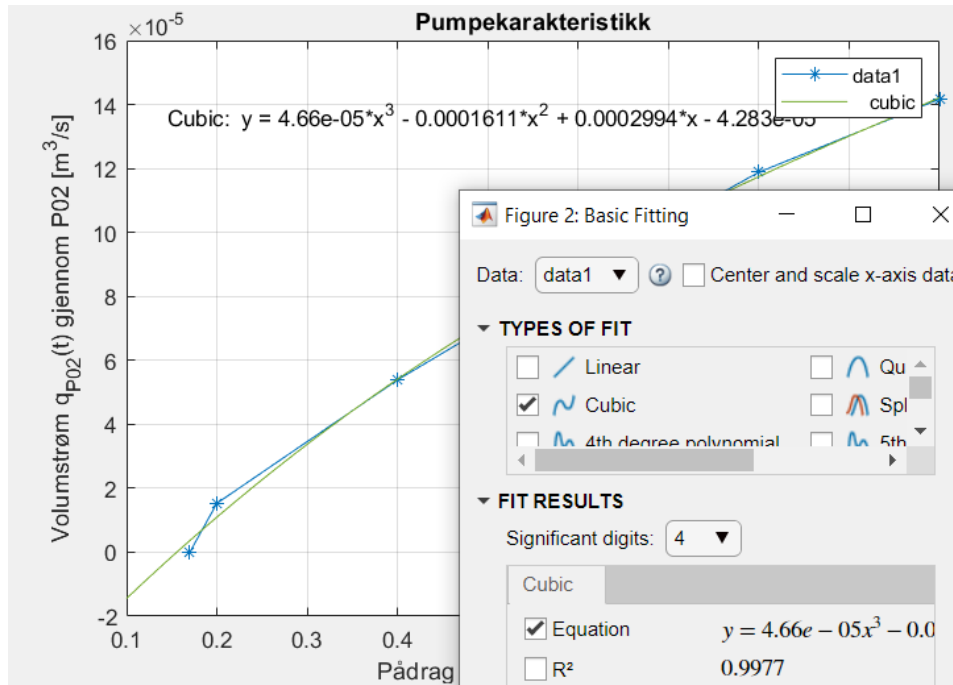
Figur 3.2: Pumpekarakteristikk for P01, P02 og P03.

Vi beregnet pumpekarakteristikken og plottet resultatet ved hjelp av Matlab. Kodeutdrag 3.1 viser hvordan dette ble utført. I linje 20 av kodeutdraget ser man at volumstrømmen til pumpen blir beregnet ved å dele volumet ($0,00924 m^3$) til full tank på tiden det tok å fylle tanken. For å få en modell i Simulink ble det utført en kurvetilpasning i Matlab. Kurvetilpasningen ble utført i plottet ved hjelp av Matlab verktøyet **Basic Fitting** som er vist i figur 3.3.

Kode 3.1: Kodeutdrag fra vedlegg H.1 for pumpekarakteristikk til P01.

```
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16 % Pumpekarakteristikk P01
17 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
18 u_P01 = [0.17, 0.2, 0.4, 0.6, 0.8, 1];
19
20 q_P01 = [0, V/639.88, V/175.71, V/103.83, V/76.54, V/64.15];
21 % Volumstrøm q_{P01} = volum full tank (V) / tid fra tom til full tank
22
23 figure
24 plot(u_P01, q_P01, '*-')
25 title('Pumpekarakteristikk')
26 xlabel('Pådrag u_{P01}(t) til pumpe P01')
27 ylabel('Volumstrøm q_{P01}(t) gjennom P01 [m^3/s]')
```

3.1 Pumpekarakteristikk

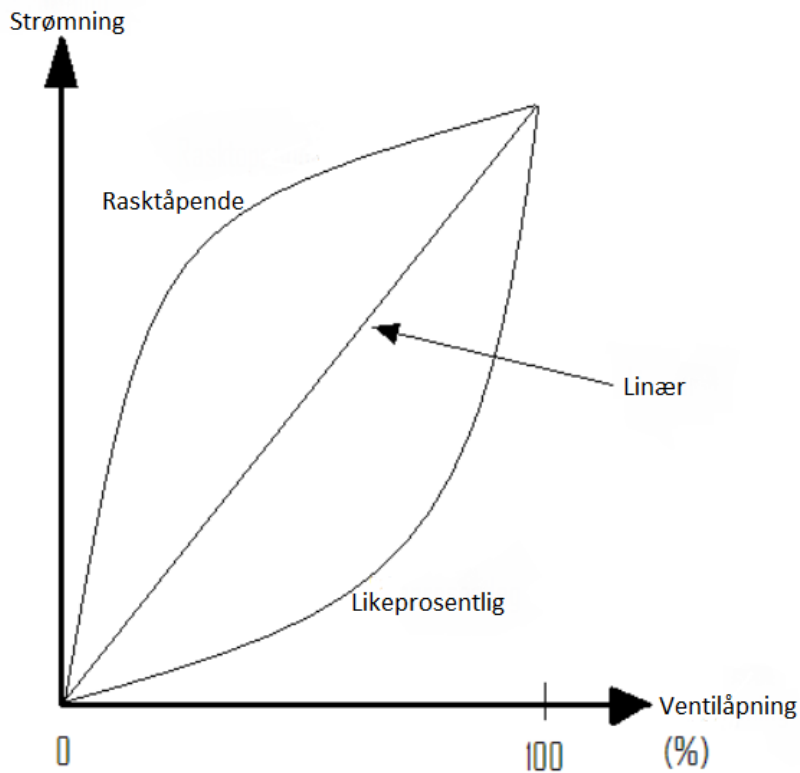


Figur 3.3: Basic fitting verktøy som ble brukt til kurvetilpassing

3.2 Ventilkarakteristikk

3.2 Ventilkarakteristikk

For å simulere ventilene beregnet vi ventilkarakteristikk. Figur 3.5 viser typiske ventilkarakteristikker.



Figur 3.4: Ventilkarakteristikker [6]

Ventilåpningen beregnet vi ved å bruke ventilligningen (3.1), hvor ventilkonstanten, K_v , blir omgjort til SI enheter for å forenkle beregningene.

3.2 Ventilkarakteristikk

Ventilligningen:

$$q(t) = K_v \cdot f(z(t)) \cdot \sqrt{\Delta p(t)} \quad (3.1)$$

- K_v er ventilkonstanten [$\frac{m^3/s}{\sqrt{Pa}}$]
- $z(t)$ er ventilåpningen [-]
- $f(z(t))$ er ventilkarakteristikken
- $\Delta p(t)$ er trykkfallet over ventilen [Pa]
- $q(t)$ er volumstrøm [$\frac{m^3}{s}$]

Vi beregnet ventilkarakteristikken ved å utføre følgende steg:

- Lagde en enkel regulator for pumpene som regulerte nivået i tankene.
- Åpnet ventilen og ventet til nivået i tanken ble stabilt.
- Leste av pumpepådraget og fant volumstrømmen i pumpekarakteristikken.
- Beregnet ventilåpningen ved hjelp av volumstrømmen.

Det ble anvendt en PI regulator for å regulere pumpene. Regulatorverdiene til pumpene er vist i ligning 3.2. Verdiene ble funnet ved å bruke Skogestads metode som er beskrevet i tabell 3.1. Regulatorverdiene var tilnærmet like for alle pumpene, brukte derfor verdiene fra ligning 3.2 til pumpe P01, P02 og P03.

Tabell 3.1: Skogestad's formel for PID tuning. Standard verdi for k_1 er 4, men en mindre verdi kan gi raskere kompensasjon mot forstyrrelse. For andre orden er T_1 den største og T_2 den minste tidskonstanten. (min betyr minste verdi) [1]

$H_p(s)$	K_p	$T_i = \frac{K_p}{K_i}$	T_d
$\frac{K}{s} e^{-\tau s}$	$\frac{1}{K(T_C + \tau)}$	$k_1(T_C + \tau)$	0
$\frac{K}{Ts+1} e^{-\tau s}$	$\frac{T}{K(T_C + \tau)}$	$\min[T, k_1(T_C + \tau)]$	0
$\frac{K}{(Ts+1)s} e^{-\tau s}$	$\frac{1}{K(T_C + \tau)}$	$k_1(T_C + \tau)$	T
$\frac{K}{(T_1s+1)(T_2s+1)} e^{-\tau s}$	$\frac{T_1}{K(T_C + \tau)}$	$\min[T_1, k_1(T_C + \tau)]$	T_2
$\frac{K}{s^2} e^{-\tau s}$	$\frac{1}{4K(T_C + \tau)^2}$	$4(T_C + \tau)$	$4(T_C + \tau)$

3.2 Ventilkarakteristikk

Regulator verdier til pumpene:

$$\begin{aligned}K &= \frac{\Delta y}{\Delta u} = \frac{h_{max}/s}{1} = \frac{0.6}{64.15} \approx 0.00935 \\T_C &= 7.2 \\K_p &= 14.85 \\T_i &= 28.8 \\K_i &\approx 0.52\end{aligned}\tag{3.2}$$

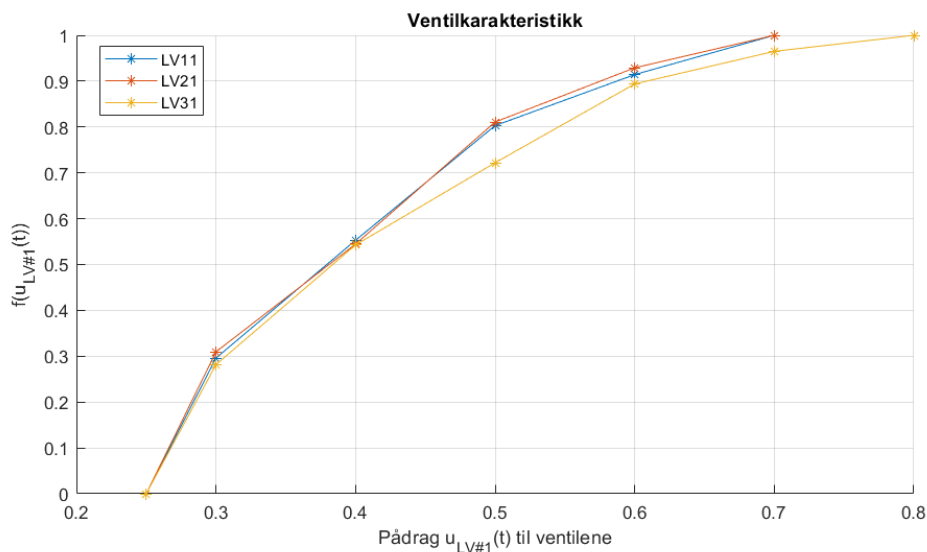
- K er maksimal forsterkning
- Δy er endringen i utgangen (volumstrøm)
- Δu er endring i inngang (pådrag)
- T_C er ønsket respons tid til systemet
- K_p er den proporsjonale forsterkningen
- T_i er integraltid
- K_i er integral forsterkning

Ventilkarakteristikken ble beregnet og plottet ved hjelp av Matlab som vist i kodeutdrag 3.2. Se figur 3.5, for plott av karakteristikken til ventil LV11, LV21 og LV31. Brukte Matlab verktøyet **Basic Fitting** for å finne en funksjon som matematisk beskriver kurven til ventilkarakteristikken.

Kode 3.2: Kodeutdrag fra vedlegg H.1 for ventilkarakteristikk til LV11.

```
43 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
44 % Ventilkarakteristikk LV11
45 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
46 u_LV11 = [0.25, 0.3, 0.4, 0.5, 0.6, 0.7];
47
48 q_LV11_maalt = [0, 2.208*10^(-5), 4.125*10^(-5), 5.989*10^(-5), ...
49 6.819*10^(-5), 7.458*10^(-5)];
50 q_LV11_max = q_LV11_maalt(1,6);
51
52 Kv_LV11 = q_LV11_max/sqrt(p); % ventilkonstant LV11 [m^3/(s*sqrt(Pa))]
53
54 f_LV11_beregnet = q_LV11_maalt/(Kv_LV11*sqrt(p));
55
56 figure
57 plot(u_LV11, f_LV11_beregnet, '*-')
58 title('Ventilkarakteristikk')
59 xlabel('P   drag u_{LV11}(t) til ventil LV11')
60 ylabel('f(u_{LV11}(t))')
```

3.2 Ventilkarakteristikk



Figur 3.5: Ventilkarakteristikk til ventilene LV11, LV21 og LV31

Det bestemmes at de manuelle ventilene, LV12, LV22 og LV32 enten er lukket eller har full åpning, og at volumstrømmen er lik på grunn av lik montering. Karakteristikken for disse er vist i kodeutdrag 3.3.

Kode 3.3: Kodeutdrag fra vedlegg H.1 for ventilkarakteristikk for de manuelle ventilene.

```

100 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
101 % Ventilkarakteristikk Manuelle (LV12, LV22, LV32)
102 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
103 u_LV12 = [0,1];
104 u_LV22 = [0,1];
105 u_LV32 = [0,1];
106
107 q_LV12_maalt = [0, 8.904*10^(-5)];
108 q_LV22_maalt = [0, 8.904*10^(-5)];
109 q_LV32_maalt = [0, 8.904*10^(-5)];
110 q_LV12_max = q_LV12_maalt(1,2);
111 q_LV22_max = q_LV22_maalt(1,2);
112 q_LV32_max = q_LV32_maalt(1,2);
113
114 Kv_LV12 = q_LV12_max/sqrt(p); % ventilkonstant LV12 [m^3/(s*sqrt(Pa))]
115 Kv_LV22 = q_LV22_max/sqrt(p); % ventilkonstant LV22 [m^3/(s*sqrt(Pa))]
116 Kv_LV32 = q_LV32_max/sqrt(p); % ventilkonstant LV32 [m^3/(s*sqrt(Pa))]

```

3.3 Modellering

3.3 Modellering

For å modellere systemet brukes formelen for massebalansen 3.3.

$$\frac{dm(t)}{dt} = \rho \cdot q_i(t) - \rho \cdot q_u(t) \quad (3.3)$$

Ved å bruke likning 3.4 i likning 3.3 får vi høyde i massebalansen.

$$m(t) = \rho \cdot A \cdot h(t) \quad (3.4)$$

Snur uttrykket med hensyn på $h(t)$ å får uttrykket vist i likning 3.5.

$$\frac{dh(t)}{dt} = \frac{1}{A} \cdot (q_i(t) - q_u(t)) \quad (3.5)$$

- $m(t)$ er massen over tid [kg]
- ρ er tettheten til vann [$\frac{N}{m^2}$]
- $q_i(t)$ er volumstrøm inn [$\frac{m^3}{s}$]
- $q_u(t)$ er volumstrøm ut [$\frac{m^3}{s}$]
- $h(t)$ er høyden over tid [m]
- A er tverrsnittetarealet til tanken [m^2]

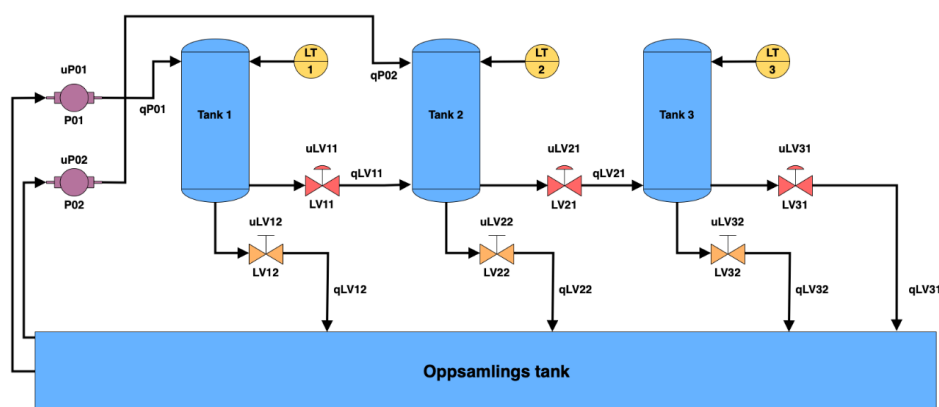
Ved å anvende ligning (3.5) utledes ligningene (3.6). For å unngå negativ verdi under kvadratrotten ble det brukt absoluttverdi. Ved å bruke absoluttverdi blir volumstrømmen alltid positiv. Derfor blir Matlab operatoren **sign** brukt for å avgjøre hvilken retning væsken strømmer gjennom ventilen. **Sign** operatoren setter verdien foran kvadratrotten til 1 dersom trykkdifferansen mellom tankene er positiv og -1 dersom trykkdifferansen er negativ.

3.3 Modellering

Matematisk modell av systemet:

$$\begin{aligned}
 \frac{dh_1(t)}{dt} &= \frac{1}{A} \cdot (q_{P01}(t) - q_{LV11}(t) - q_{LV12}(t)) \\
 \frac{dh_2(t)}{dt} &= \frac{1}{A} \cdot (q_{P02}(t) + q_{LV11}(t) - q_{LV21}(t) - q_{LV22}(t)) \\
 \frac{dh_3(t)}{dt} &= \frac{1}{A} \cdot (q_{LV21}(t) - q_{LV31}(t) - q_{LV32}(t)) \\
 q_{LV11}(t) &= K_{v,LV11} \cdot f(z_{LV11}(t)) \cdot \text{sign}(p_1 - p_2) \cdot \sqrt{|p_1(t) - p_2(t)|} \\
 q_{LV12}(t) &= K_{v,LV12} \cdot f(z_{LV12}(t)) \cdot \sqrt{p_1(t)} \\
 q_{LV21}(t) &= K_{v,LV21} \cdot f(z_{LV21}(t)) \cdot \text{sign}(p_2 - p_3) \cdot \sqrt{|p_2(t) - p_3(t)|} \\
 q_{LV22}(t) &= K_{v,LV22} \cdot f(z_{LV22}(t)) \cdot \sqrt{p_2(t)} \\
 q_{LV31}(t) &= K_{v,LV31} \cdot f(z_{LV31}(t)) \cdot \sqrt{p_3(t)} \\
 q_{LV32}(t) &= K_{v,LV32} \cdot f(z_{LV32}(t)) \cdot \sqrt{p_3(t)}
 \end{aligned} \tag{3.6}$$

- $q_{P0\#}(t)$ volumstrøm til pumpene
- $q_{LV\#\#}(t)$ volumstrøm til ventilene
- $K_{v,LV\#\#}(t)$ ventilkonstanter
- $f(z_{LV\#\#}(t))$ ventilåpninger
- $p_{\#}(t)$ trykk i tankene

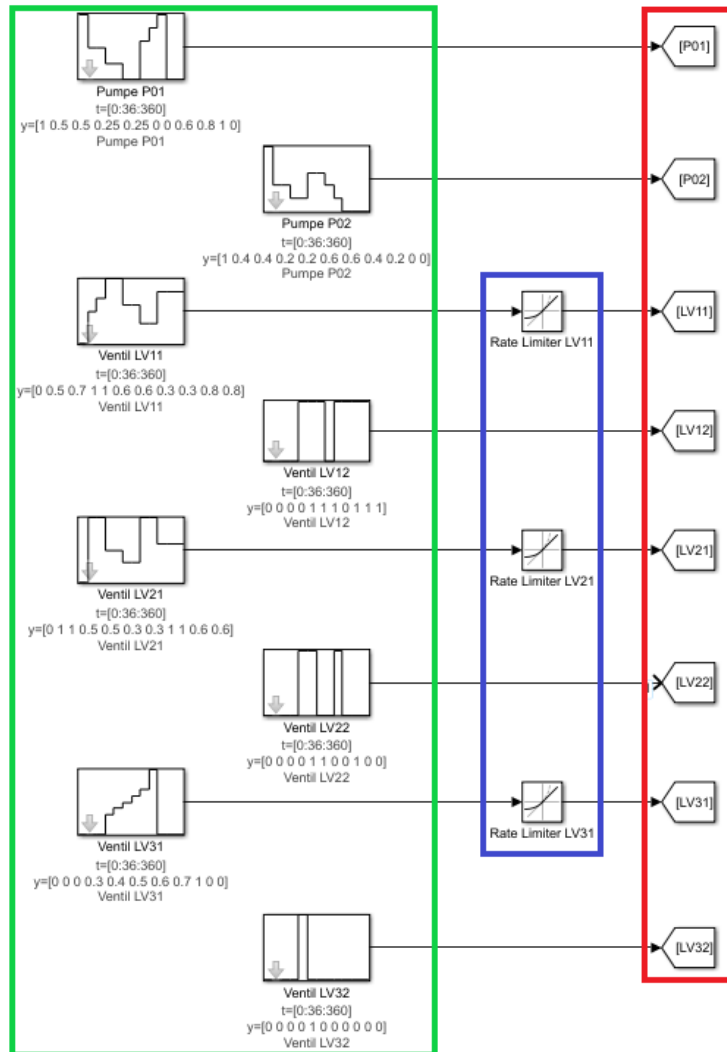


Figur 3.6: Modell av tretank anlegget.

3.4 Simulinkmodell

Figur 3.7 og 3.8 viser Simulinkmodellen som ble brukt for å verifisere at den matematiske modellen samsvarer med det fysiske tretankanlegget. Ved å tilføre forskjellig pådrag (grønn firkant, figur 3.7) på inngangene i **Matlab funksjonen**, som er vist i figur 3.8, kan man se den teoretiske høyden til tankene i scopet. Pådragene til ventilene blir sendt igjennom en **Rate Limiter** som simulerer åpningshastigheten til de automatiske ventilene. Alle pådragene blir videreført til **Matlab funksjonen** ved hjelp av **GoTo** (rød firkant, figur 3.7) og **From** blokkene (oransje firkant, figur 3.8). I **Matlab funksjonen** blir pådragene behandlet av den matematiske modellen (kode for Matlab funksjon se vedlegg H.2). **Matlab funksjonen** beregner den derivert høyden og videreføres til **integratorene** (lilla firkant, figur 3.8). Resultatet av den aktuelle høyden til hver tank blir vist i scopet. Etter utgangsvariablene blir integrert blir de så returnert inn på inngangside av **Matlab funksjonen** via **GoTo** (rød firkant, figur 3.8) til **From** blokkene, ved å gjøre dette kan modellen alltid ta hensyn til den nåværende høyden i de forskjellige tankene.

3.4 Simulinkmodell



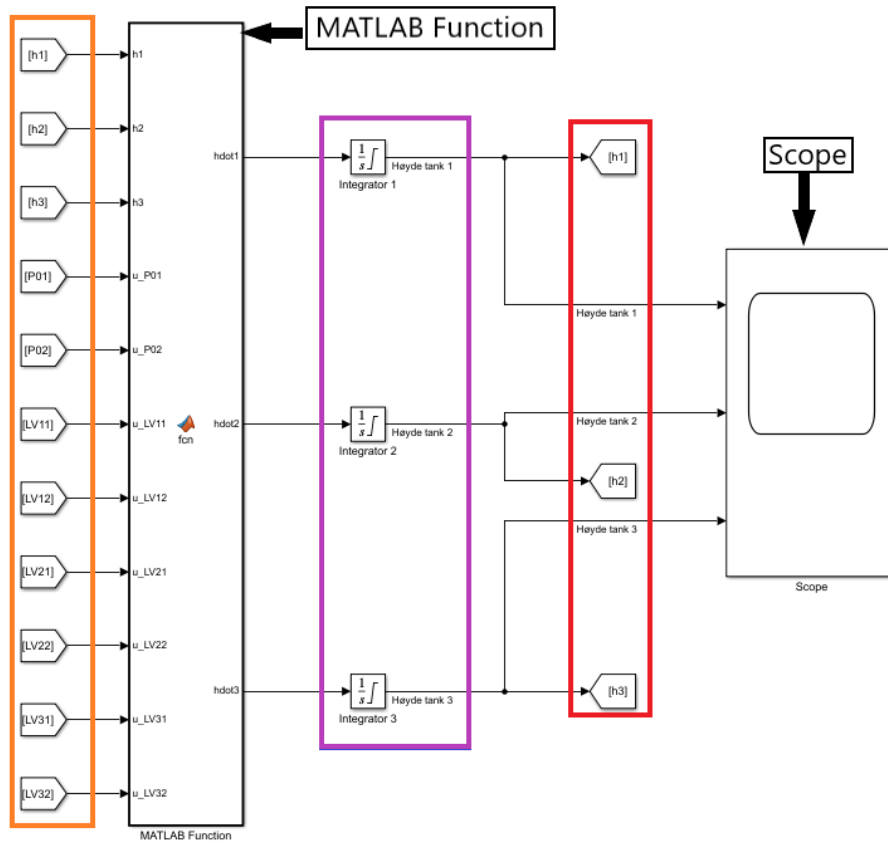
Figur 3.7: Oversikt over pådragsseksjon til simulinkmodellen.

Grønn firkant: Multiple step block (Her defineres pådraget til automatiske/manuelle ventilene og pumpene).

Blå firkant: Rate limiter til de automatiske ventilene (LV11, LV21, LV31). Styrer åpningshastigheten til ventilene.

Rød firkant: GoTo blokk (Sender signalet videre til den korresponderende From blokk).

3.5 Verifisering



Figur 3.8: Oversikt over prosess seksjonen til simulinkmodellen.

Oransj firkant: From blokk (Tar imot signalet fra den korresponderende GoTo blokken).

MATLAB Function

Lilla firkant: Integrator.

Rød firkant: GoTo (Sender signalet videre til den korresponderende From blokk).

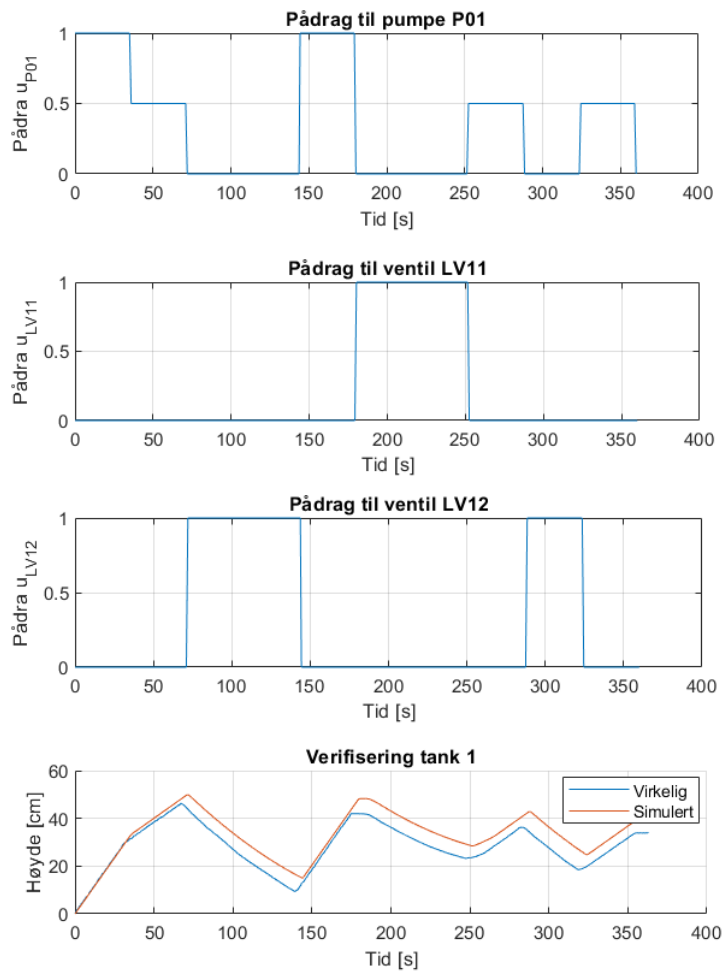
Scope

3.5 Verifisering

For å verifisere dynamikken til tankene ble systemet simulert i Simulink med forskjellig pådrag på pumpene og ventilene. Ut fra PLSen hentet vi ut en excel-fil som vi importerte inn i Matlab via SD kortet fra HMI skjermen

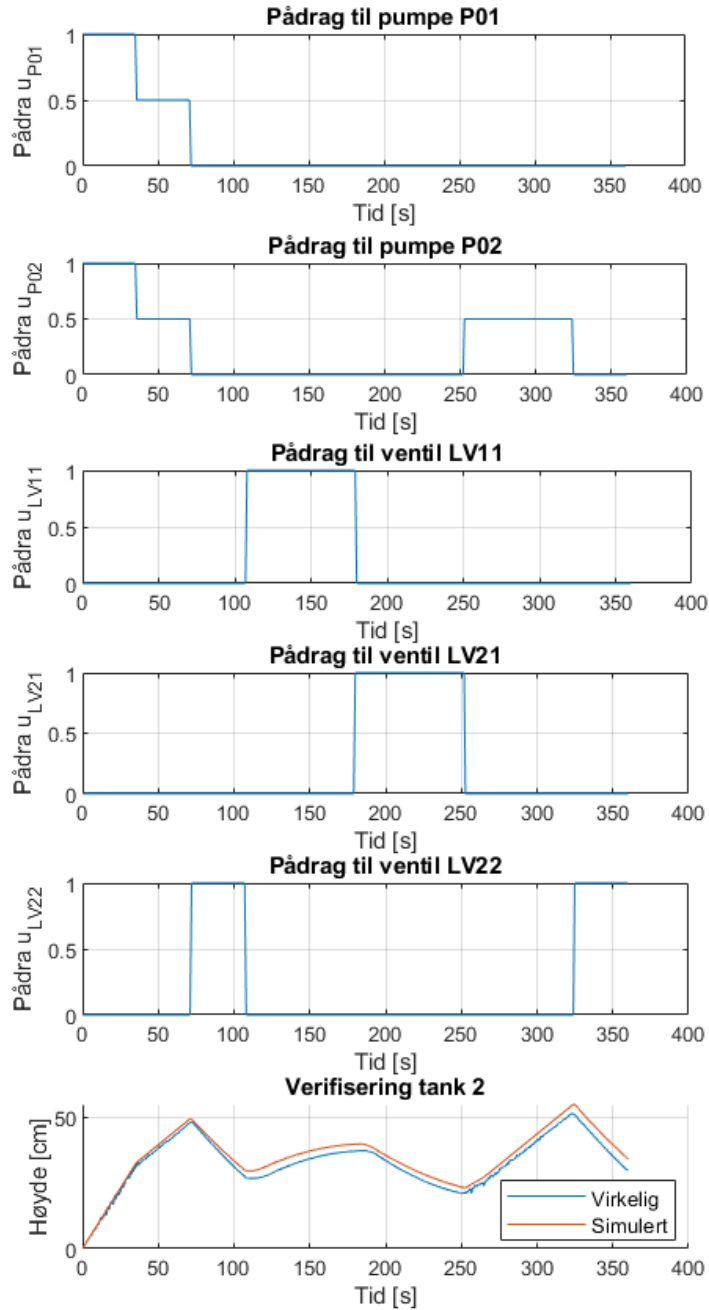
3.5 Verifisering

for å sammenligne virkelig og simulert respons. I figur 3.9 viser de tre første grafene pådragene til pumpe P01, ventil LV11 og LV12. Dynamikken til det teoretiske (Simulink) og det fysiske (tretankanlegget) systemet vises i den fjerde grafen. Her kan man se at den teoretiske og den virkelige dynamikken samsvarer fint. Vi brukte samme fremgangsmåte på tank 2 og tank 3, se figur 3.10 og 3.11 hvor vi verifiserte at den teoretiske modellen stemmer med det fysiske tretankanlegget. Kode for hvordan verifiseringene ble fremstilt kan ses i vedlegg H.3.



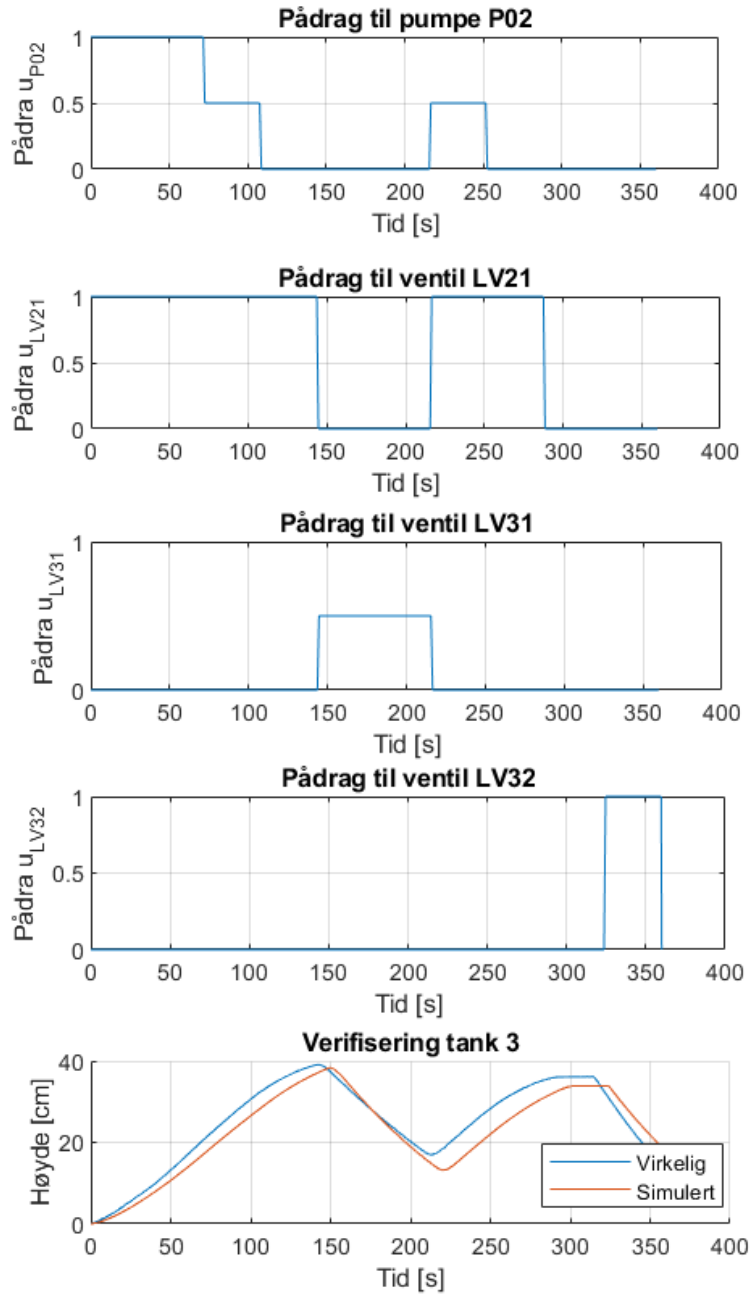
Figur 3.9: Verifisering av tank 1

3.5 Verifisering



Figur 3.10: Verifisering av tank 2

3.5 Verifisering



Figur 3.11: Verifisering av tank 3

Kapittel 4

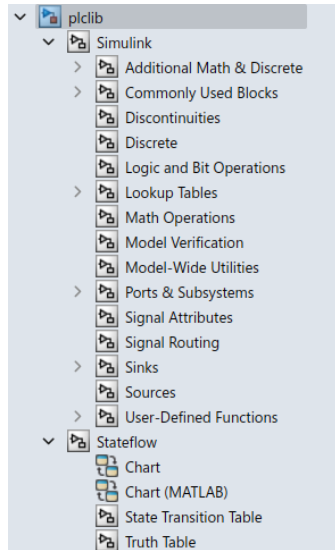
PLC Coder og Sysmac Studio

4.1 PLC Coder

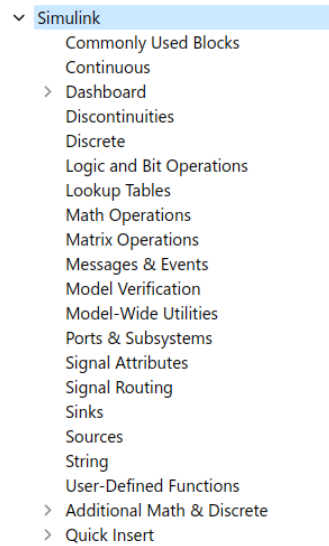
Simulink PLC Coder genererer strukturert tekst (ST) og ladder diagram (LD) fra Simulink modeller, Matlab funksjoner og Stateflow charts som støttes av **plclib** [5] i Simulink. Biblioteket **plclib** er et utdrag fra det originale Simulink biblioteket. Forskjellen mellom bibliotekene er at **plclib** ikke inneholder kontinuerlige blokker og funksjoner på grunn av at PLSene opererer med diskrete variabler. Et utdrag av bibliotekene er vist i figur 4.1 og 4.2. Strukturert tekst støttes av Omron Sysmac Studio, SIEMENS TIA Portal og 3S-Smart Software Solutions CODESYS. Det er mulig å produsere ladder diagram direkte i Simulink, men det er mer krevende sammenliknet med å benytte seg av LD direkte i PLSens IDE. Det er kun Rockwell Automation Studio 5000 som støtter både strukturert tekst og ladder diagram fra Simulink. Strukturert tekst blir generert i PLCopen XML og andre filformater som støttes av fler integrerte utviklingsmiljøer (IDE). Simulink PLC Coder følger IEC 61131-3 standarden og kan genererer testbenker som hjelper med å verifisere den strukturerte teksten og ladder diagram koden som blir generert. En testbenk er et Matlab script som blir brukt til å teste en algoritme eller kode i din designede matlabfunksjon eller modell. Testbenken tester om koden er numerisk lik og har samme funksjonalitet som utgangsverdiene til simulinkmodellen. Vi har ikke brukt testbenkene grunnet lett oppkobling til fysisk system og kunne testet den genererte koden

4.1 PLC Coder

direkte. For mer info om testbenk i matlab se referanse [2] og [3].



Figur 4.1: Figuren viser strukturen og en overordnet liste av innholdet i `plclib`.



Figur 4.2: Figuren viser strukturen og en overordnet liste av innholdet i Simulink biblioteket.

4.1.1 Generering av kode til Sysmac Studio

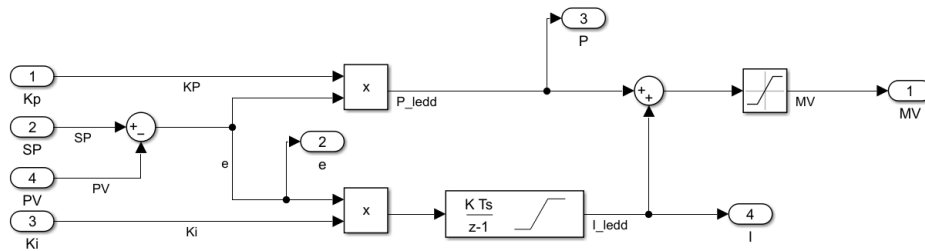
For å vise funksjonaliteten benytter vi et eksempel hvor vi generere strukturert tekst fra et subsystem i Simulink til Omron Sysmac Studio. Under ser man en stegvis gjennomgang av hvordan man genererer strukturert tekst fra et subsystem i simulink. Eksempelet brukt under er generering av vår pumpe kontroller. Likning 4.1 viser matematisk hvordan vi har konstruert vår regulator.

$$u_{P01} = MV = K_p \cdot e + K_i \cdot T_s \int_0^t e dt \quad (4.1)$$

- MV - manipulated value
- e - avvik
- Ts - tidsskritt

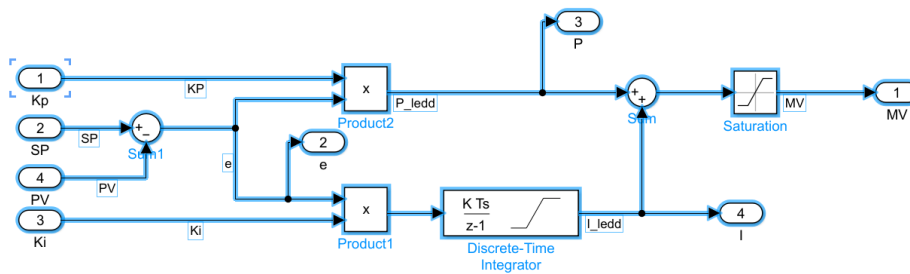
4.1 PLC Coder

1) Definer og lag en Simulink modell som du ønsker å generere en kode ifra. Simulink PLC Coder er kompatibel med blokkene i **plclib** for generering av kode.



Figur 4.3: Blokkene som ble brukt for å konstruere kontrolleren støttes av biblioteket **plclib**. For å få utgangsvariablene som ønsket i den genererte koden måtte de defineres som utgangsporter i Simulink. Verdiene til K_p , K_i og SP ble ikke satt som konstanter, men inngangsporter for at de skal kunne justeres via HMI skjermen.

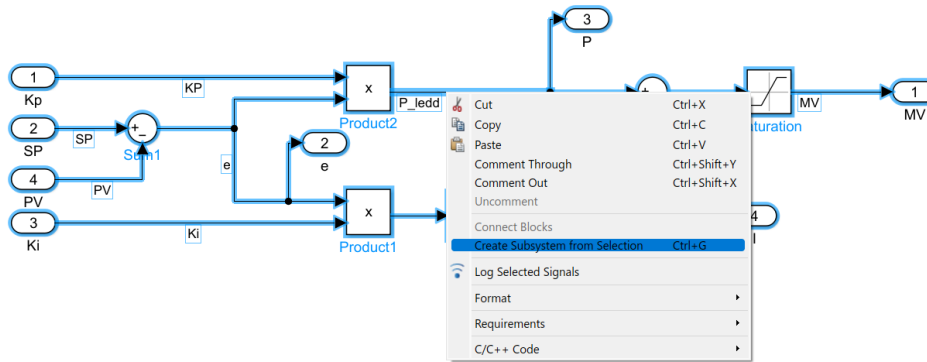
2) Identifiser modellens komponenter som du ønsker å generere en kode for som kan importeres til en PLS.



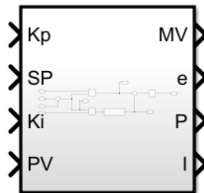
Figur 4.4: Markerte blokkene vi ønsket å konvertere til et subsystem.

4.1 PLC Coder

3) Generer subsystem fra de markerte blokkene.



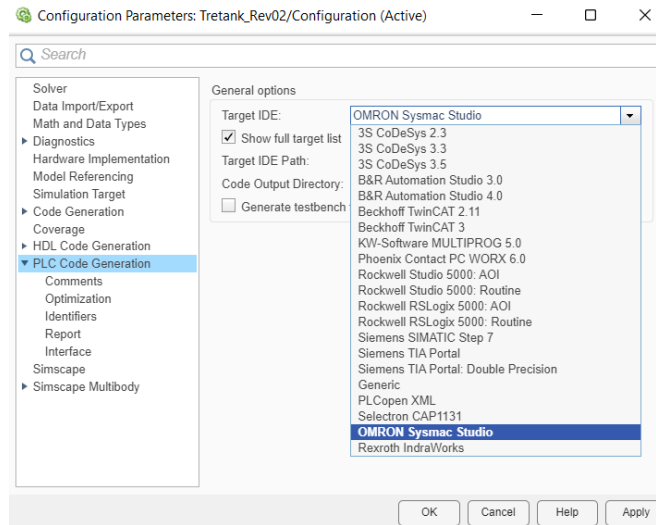
Figur 4.5: Høyreklikket på de markerte blokkene og valgte alternativet **Create Subsystem from Selection**.



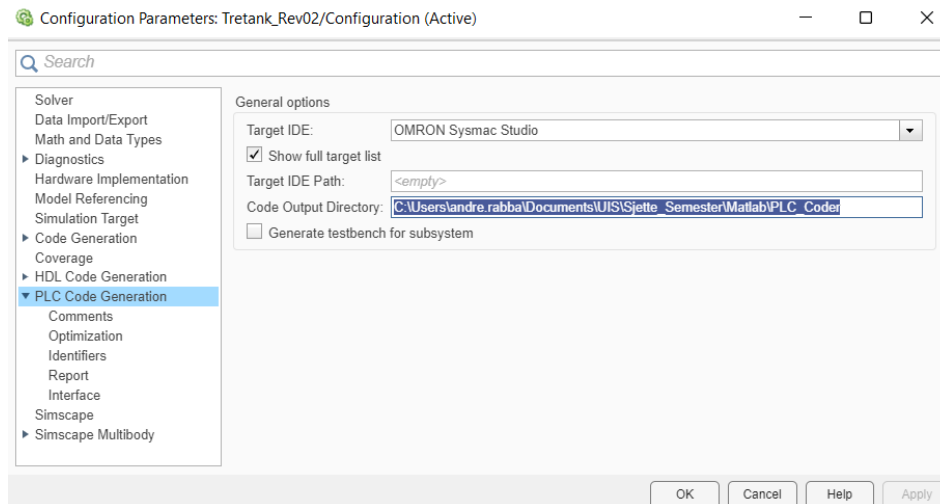
Figur 4.6: Ferdig generert subsystem fra blokkdiagram.

4.1 PLC Coder

4) Identifiser og velg den ønskede Integrated Development Environment (IDE) til PLSen og hvor du ønsker å lagre koden på din PC/Mac.



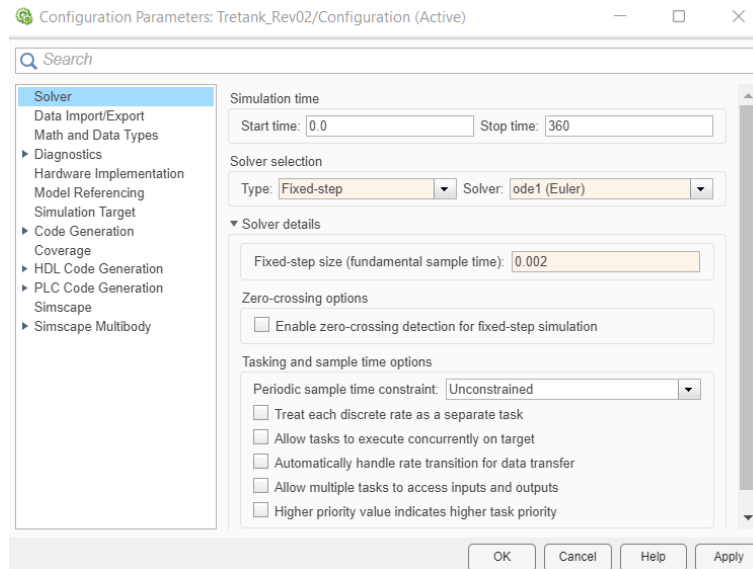
Figur 4.7: Gikk til fanen **Modelling** i Simulink og trykket på **Model Settings**. Valgte **PLC Code Generation** → **Target IDE** → **Omron Sysmac Studio** fra rullegardinmenyen.



Figur 4.8: Under **Code Output Directory** valgte vi stien vi ønsket å lagre filen på. Trykket **Apply** → **OK**. Den genererte filen ble lagret som både **.st** og **.xml**-fil.

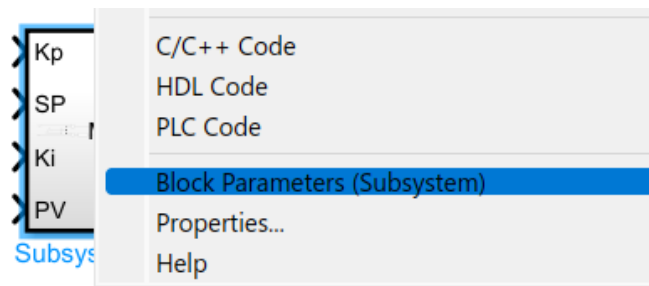
4.1 PLC Coder

5) Velg ønsket **solver** i forhold til hvordan IDEen er konfigurert.



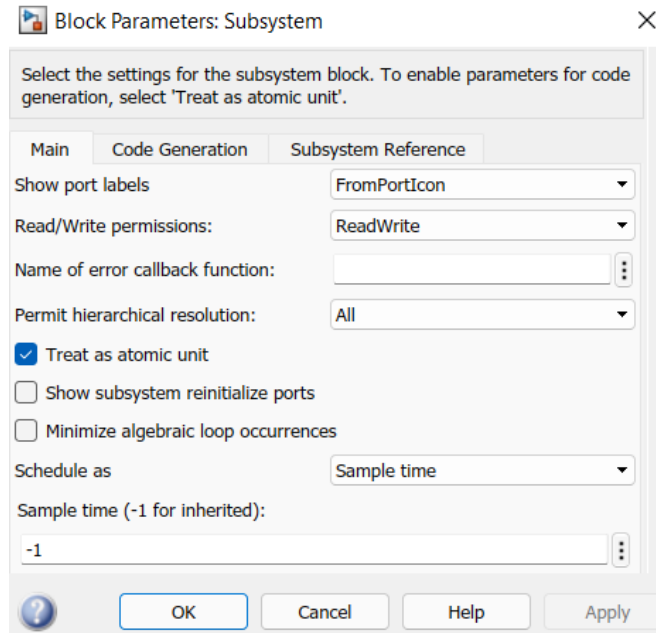
Figur 4.9: Gikk inn på **Model Settings**, valgte **Solver**. I **Solver selection** ble det valgt **Type: Fixed-step** og **Solver: ode1(Euler)** for vårt system. Under **Solver details** ble **Fixed-step size (fundamental sample time)** satt til 0.002 sekunder på grunn av steglengden til Omron PLSen som vi brukte. Trykket **Apply** → **OK**.

6) Konfigurer subsystemet til å være **atomic**. Det betyr at Simulink skal prioritere og kjøre hele koden til subsystemet uten at det blir påvirket av andre deler av simulinkmodellen.



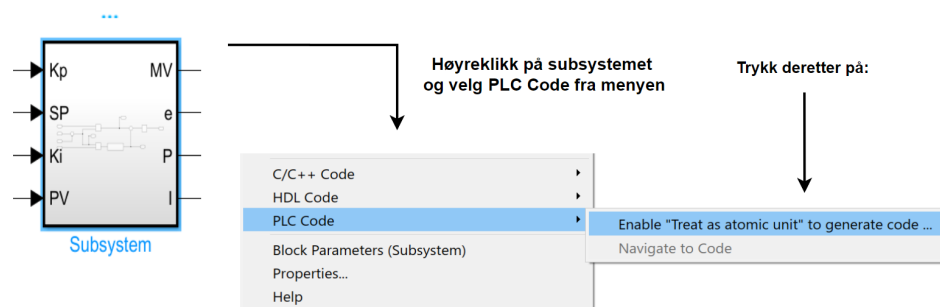
Figur 4.10: Høyreklikket på subsystemet, valgte **Block Parameter (Subsystem)** fra rullegardinmenyen.

4.1 PLC Coder



Figur 4.11: Trykket på fanen **Main** og huket av ruten **Treat as atomic unit**. Trykket **Apply** og **OK**.

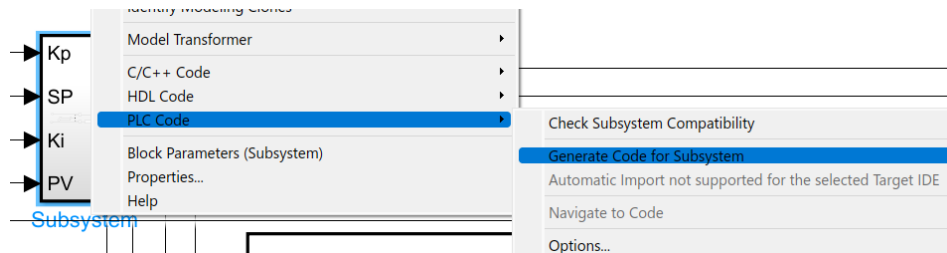
Alternativt kan man gjøre det ved å høyreklikke på subsystemet og følge stegene i figur 4.12. Dette er kun mulig å utføre dersom subsystemet ikke allerede er en **atomic unit**.



Figur 4.12: Høyreklikket på subsystemet, valgte **PLC Code** → **Enable "Treat as atomic unit" to generate code ...**

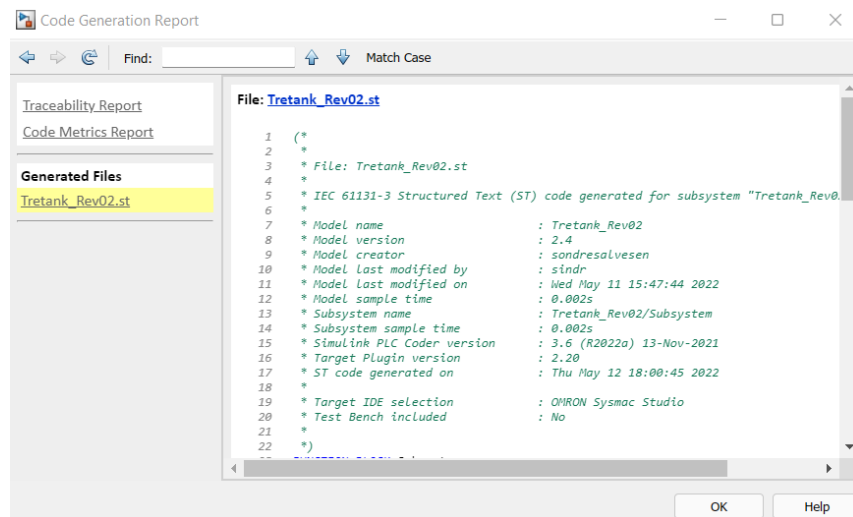
4.1 PLC Coder

- 7) Simuler modellen for å verifisere at modellen fortsatt fungerer som tiltenkt.
- 8) Generere strukturert tekst fra subsystemet.



Figur 4.13: Vi genererte strukturert tekst ved å høyreklikke på subsystemet. Valgte **PLC Code** → **Generate Code for Subsystem** fra rullegardinmenyen.

- 9) Inspiser den genererte koden.



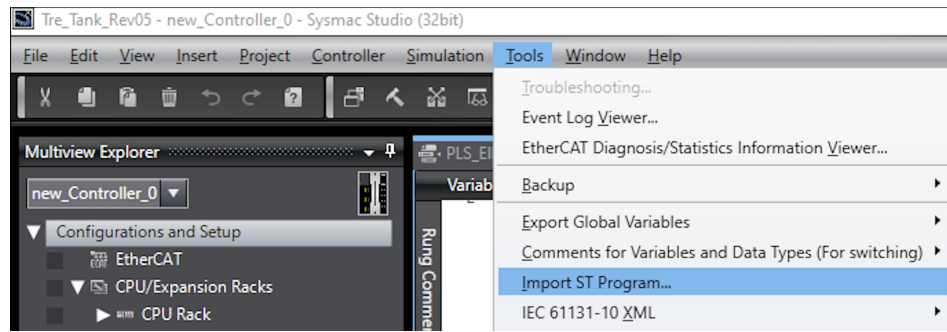
Figur 4.14: Etter at vi hadde generert koden fikk vi opp vinduet **Code Generation Report**. Under **Generatet Files** i menyen til venstre fant vi den strukturerte tekst filen til pumpe regulatoren. Den enkle PI regulatoren genererte 79 linjer med kode, inkludert kommentarer. For fullstendig kode se vedlegg H.4.

4.1 PLC Coder

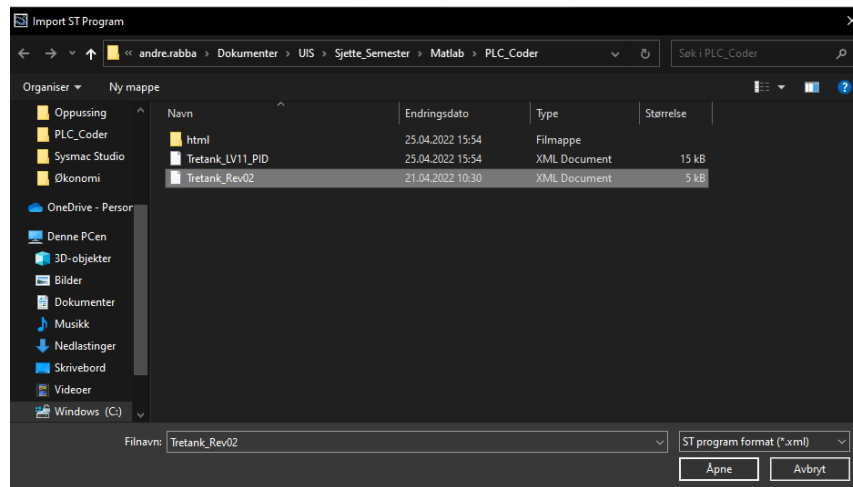
4.1.2 Importere strukturert tekst fra Simulink inn i Sysmac Studio

For å importere den genererte XML-filen fra Simulink inn i Sysmac Studio som en funksjonsblokk utførte vi stegene under.

- 1) Åpne ønsket prosjekt hvor den strukturerte teksten skal implementeres.
- 2) Importer den strukturerte teksten som ble generert av Simulink.



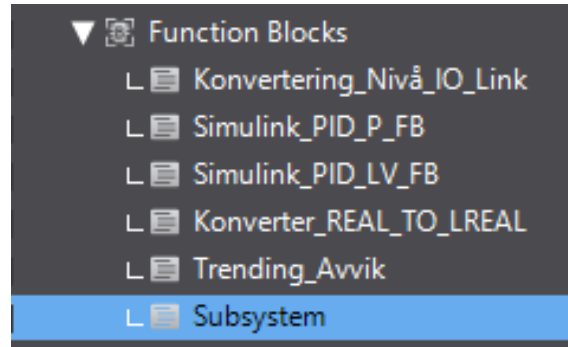
Figur 4.15: I hovedfanen trykket vi på **Tools** → **Import ST Program...**



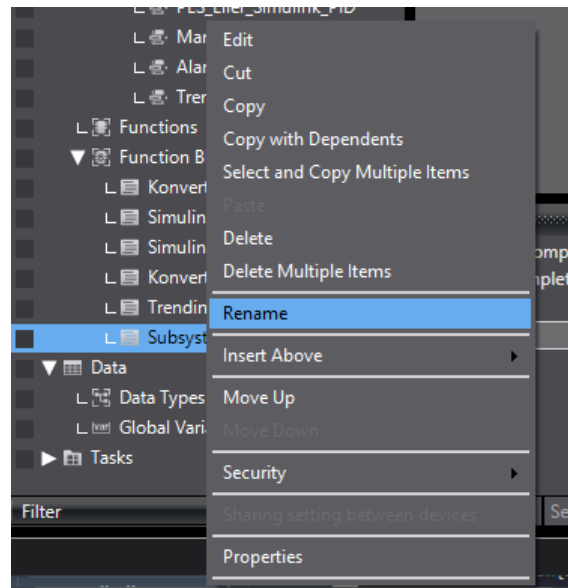
Figur 4.16: Valgte ST filen som skal importeres.

4.1 PLC Coder

Filen blir importert inn i Sysmac Studio med navnet **Subsystem**.



Figur 4.17: Den importerte ST filen ble plassert under stien **Functions Blocks** i Sysmac Studio.



Figur 4.18: For å organisere de importerte funksjonsblokkene ble navnet til funksjonsblokken endret til et mer passende navn. Ved å høyreklikke på den nye funksjonsblokken, eller trykke hurtigtasten **F2** kan du endre navn på funksjonsblokken.

4.2 Justeringer av generert kode

4.2 Justeringer av generert kode

For at den importerte funksjonsblokken skal fungere optimalt sammen med vårt PLS program ble følgende justeringer utført.

4.2.1 Variabler

Vi opprettet to boolske variabler fordi ladder programmering krever at alle funksjonsblokker skal ha en boolsk inngangsvariabel for å aktivere blokken og en utgangsvariabel for å videresende signalet. Vi opprettet en boolsk inngangsvariabel med navn **Enable** og en boolske utgangsvariabelen med navn **Running**. Figur 4.19 viser variabellisten for inn- og utgangsvariablene etter at endringer utført.

Namespace - Using								
	Name	In/Out	Data Type	Edge	Initial Value	Retain	Constant	Comment
Internals								
In/Out								
Externals	Enable	Input	BOOL	No Edge		<input type="checkbox"/>	<input type="checkbox"/>	
	Modus	Input	SINT	No Edge		<input type="checkbox"/>	<input type="checkbox"/>	
	Ki	Input	LREAL	No Edge		<input type="checkbox"/>	<input type="checkbox"/>	
	Kp	Input	LREAL	No Edge		<input type="checkbox"/>	<input type="checkbox"/>	
	SP	Input	LREAL	No Edge		<input type="checkbox"/>	<input type="checkbox"/>	
	PV	Input	LREAL	No Edge		<input type="checkbox"/>	<input type="checkbox"/>	
	Running	Output	BOOL	No Edge		<input type="checkbox"/>	<input type="checkbox"/>	
	MV	Output	LREAL	No Edge		<input type="checkbox"/>	<input type="checkbox"/>	
	e	Output	LREAL	No Edge		<input type="checkbox"/>	<input type="checkbox"/>	
	P	Output	LREAL	No Edge		<input type="checkbox"/>	<input type="checkbox"/>	
	I	Output	LREAL	No Edge		<input type="checkbox"/>	<input type="checkbox"/>	

Figur 4.19: Variabelliste etter endringer

4.2 Justeringer av generert kode

4.2.2 Endring av kode i funksjonsblokken

For at den boolske inngangsvariabelen skal aktivere funksjonsblokken måtte vi implementere en **IF** statement. Det ble i tillegg lagt inn en kommando som setter **MV** (pådragverdien) lik 0 hvis **Metode** er satt til 0. Kodeutdraget 4.1 viser hvordan statementen og kommandoen ble implementert.

Kode 4.1: Utdrag fra ST kode i funksjonsblokken i PLS programmet. Se PLS program i vedlegg F.2.

```
1 IF Enable THEN
2   Running := TRUE;
3   CASE Modus OF
4     SINT#0:
5       (* SystemInitialize for Atomic SubSystem: '<Root>/Subsystem' *)
6       (* InitializeConditions for DiscreteIntegrator: ...
          '<S1>/Discrete-Time Integrator' *)
7       c_DiscreteTimeIntegrator_DS := LREAL#0.0;
8       MV := LREAL#0;
9       (* End of SystemInitialize for SubSystem: '<Root>/Subsystem' *)
10      ...
11      ...
12      ...
47    END_CASE;
48 ELSE
49   Running := FALSE;
50   MV := LREAL#0;
51 END_IF;
```

Kapittel 5

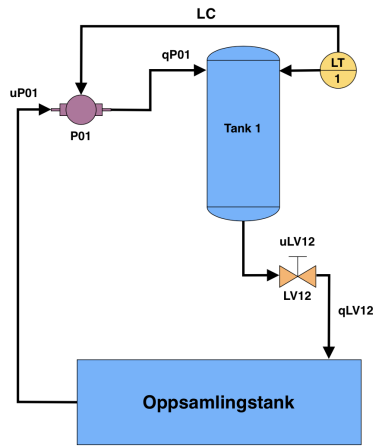
Simulering og regulering i Simulink

Vi skal i dette kapitlet se på den teoretiske (Simulink) reguleringen av pumpe P01 og den automatiske ventilen LV11.

5.1 Nivåregulering tank 1: Pumpe P01

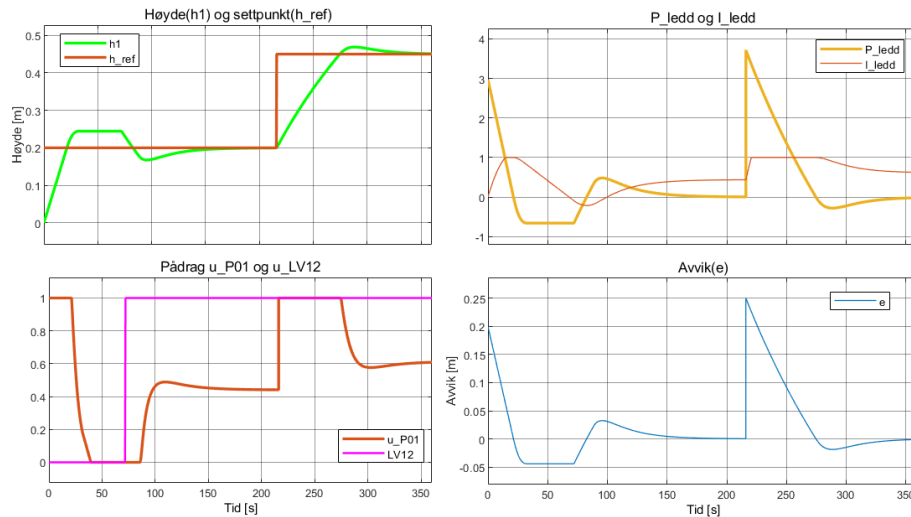
For å utføre nivåregulering av tank 1 så vi på systemet vist i figur 5.1, hvor pumpe P01 er tilførselen, nivåmåler LT1 bestemmer pådraget til pumpen og ventil LV12 simulerer forstyrrelse på systemet. På grunn av utformingen til systemet vi ønsket å regulere ble regulatoren til pumpe P01 konstruert som en integrator. For mer informasjon om beregninger og verdier se seksjon 3.2. Simuleringsresultatet fra Simulink er vist i figur 5.2 hvor regulatorens egenskaper er vist. For å eventuelt redusere oversvinget kan vi øke ønsket tidsrespons, eller redusere P-leddet.

5.1 Nivåregulering tank 1: Pumpe P01



Figur 5.1: Figuren viser delen av prosessen vi brukte til å utføre simuleringen og reguleringen.

I figur 5.2 blir simuleringresultatene presentert. Simuleringene viser kompensering- og følgeegenskapene til pumperegulatoren.

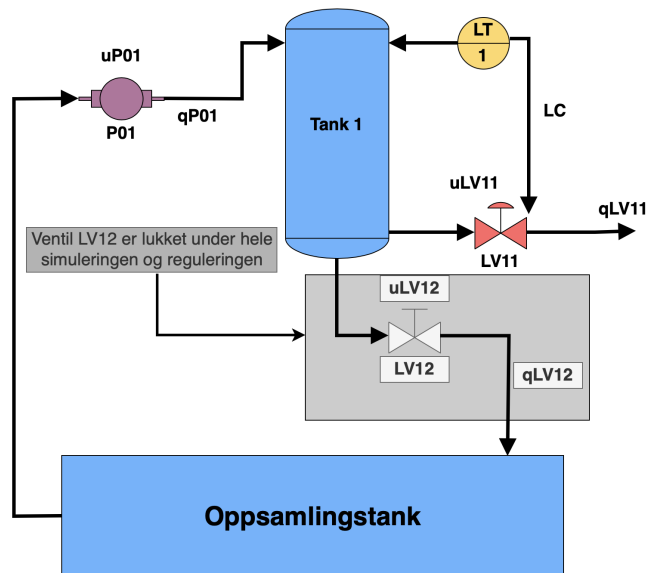


Figur 5.2: Figuren viser simuleringresultatet av høyden i tanken, P og I leddet til regulatoren, pådraget til pumpe P01 og avviket til høyden.

5.2 Nivåregulering tank 1: Ventil LV11

5.2 Nivåregulering tank 1: Ventil LV11

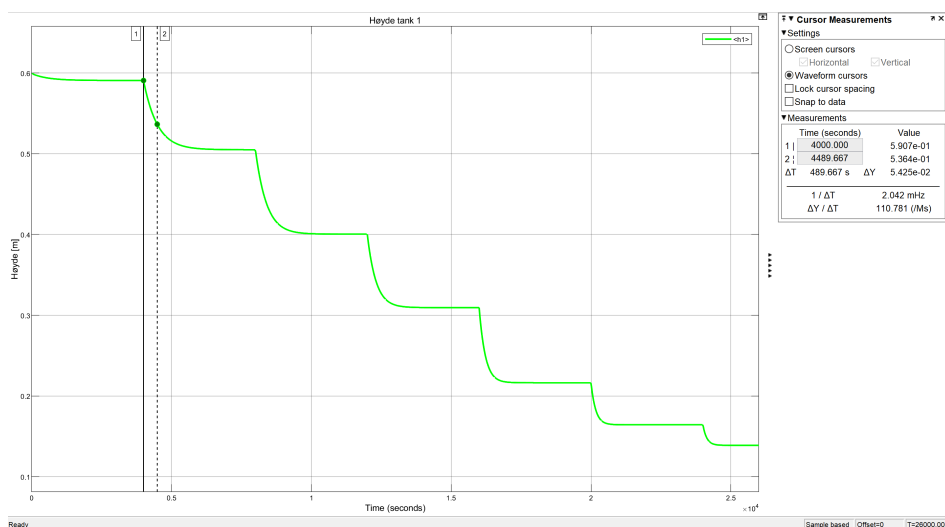
For å utføre nivåregulering av tank 1 så vi på systemet vist i figur 5.3, hvor nivåmåler LT1 bestemmer pådraget til ventil LV11 og pumpe P01 simulerer forstyrrelse på systemet. På grunn av oppbyggingen av systemet så vi det som et andre ordens system. Grunnet nivåendring oppstår det forskjellig trykk i tank 1 som fører til ulik gjennomstrømning i ventil LV11. Forskjellen medførte at vi fikk forskjellig forsterkninger(K) og tidskonstanter(T) ved ulike arbeidspunkt som førte til at det ble forskjellige K_p og K_i verdier. Vi benyttet reguleringsmetoden **Gain Scheduling** for å ta hensyn til endringen i forsterkningen og tidskonstanten. **Gain Scheduling** er en avansert reguleringsmetode som er krevende å få til i Sysmac Studio med den kunnskapen vi besitter. Ved å benytte Simulink PLC Coder viser vi nytteverdien til dette verktøyet ved å enkelt kunne konvertere PI regulatoren til ventil LV11 direkte over til strukturert tekst fra Simulinkmodellen.



Figur 5.3: Figuren viser delen av prosessen vi brukte til å utføre simuleringen og reguleringen.

5.2 Nivåregulering tank 1: Ventil LV11

For å finne forsterkningsverdiene og tidskonstantene ble det brukt et pumpepådrag på 0.3(30%) som arbeidspunkt. Verdiene ble funnet ved å sette forskjellig pådrag på ventilen og ventet til høyden i tanken ble stabil. Vi avleste simuleringsresultatene i Simulink ved hjelp av verktøyet **cursors** og beregnet verdiene ut fra resultatet. Metoden for å utføre målingene er vist i figur 5.4.



Figur 5.4: Bildet viser hvordan dynamikken til høyden responderer på et sprang i ventilåpningen. Cursors er brukt til å måle tidskonstanten som er 63.2% av endringen i høyden.

Verdiene som ble avlest fra målingene var endringen i høyde (Δh) og tidskonstanten (T). Ved hjelp av disse verdiene ble det beregnet ulike forsterkningsverdier (K) ved forskjellige sprang, formel for beregning av K er vist i likning 5.1. For å kunne enklere bestemme responstid (T_c) ble det beregnet et intervall for hvert sprang. Intervallet ble beregnet ved å gange tidskonstanten (T) med 0.2 og 0.5. Resultat av alle beregninger er oppført i tabell 5.1.

$$K = \frac{\Delta h}{\Delta u} \quad (5.1)$$

5.2 Nivåregulering tank 1: Ventil LV11

Tabell 5.1: Gain Scheduling tabell for ventil LV11

u_{LV11}	h	Δh	T	Δu	K	T_c
0,353	0,591					
0,365	0,505	-0,086	489,7	0,012	-7,17	[97,94 \rightarrow 244,85]
0,3865	0,4	-0,105	393	0,0215	-4,88	[78,6 \rightarrow 196,5]
0,418	0,309	-0,091	306	0,0315	-2,89	[61,2 \rightarrow 153]
0,4865	0,216	-0,093	217,5	0,069	-1,35	[43,5 \rightarrow 108,75]
0,5865	0,164	-0,052	163	0,1	-0,52	[32,6 \rightarrow 81,5]
0,6865	0,139	-0,025	135	0,1	-0,25	[27 \rightarrow 67,5]

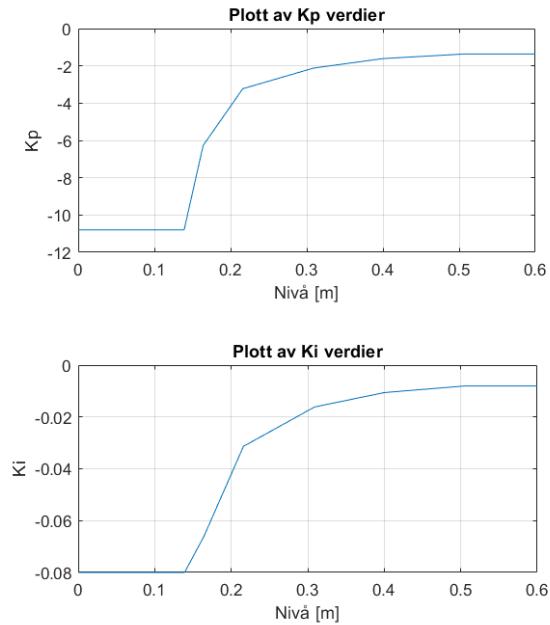
Utfra resultatene i tabell 5.1 ble Skogestad's formler for PID tuning brukt for å beregne K_p og K_i verdier. Disse beregningen ble gjort i Matlab og kan sees i kodeutdrag 5.1. I linje 175 og 180 blir første og siste verdien av vektoren gjentatt for å få riktig lengde på K_p og K_i vektoren.

Kode 5.1: Kodeutdrag fra vedlegg H.1 for beregning av PI verdier til ventil LV11.

```
160 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
161 % PI verdier LV11
162 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
163 T = [135, 163, 217.5, 306, 393, 489.7];
164  $\Delta_h$  = [-0.025, -0.052, -0.093, -0.091, -0.105, -0.086];
165  $\Delta_u$  = [0.1, 0.1, 0.069, 0.0315, 0.0215, 0.012];
166
167 for i = 1:6
168     K(i) =  $\Delta_h(i)/\Delta_u(i)$ ;
169 end
170
171 Tc = 50;
172 for i = 1:6
173     Kp(i) = T(i)/(K(i)*Tc);
174 end
175 Kp = [Kp(1), Kp, Kp(6)];
176
177 for i = 1:6
178     Ki(i) = Kp(i)/min(T(i), 4*Tc);
179 end
180 Ki = [Ki(1), Ki, Ki(6)];
181
182 h_break = [0, 0.139, 0.164, 0.216, 0.309, 0.4, 0.505, 0.6];
```

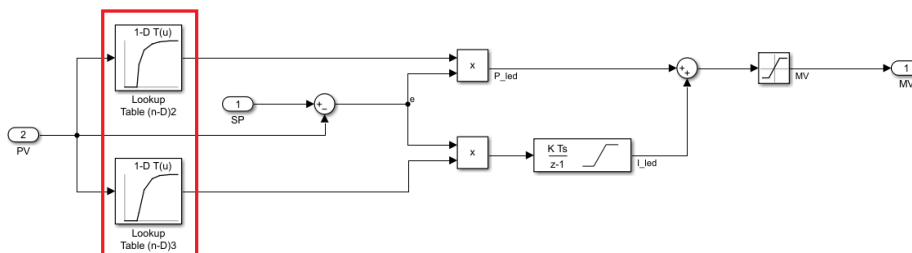

5.2 Nivåregulering tank 1: Ventil LV11

Figur 5.5 viser plott av verdiene til K_p og K_i for å vise ulinæriteten til prosessen.



Figur 5.5: Plott som viser de ulike verdiene K_p og K_i verdiene

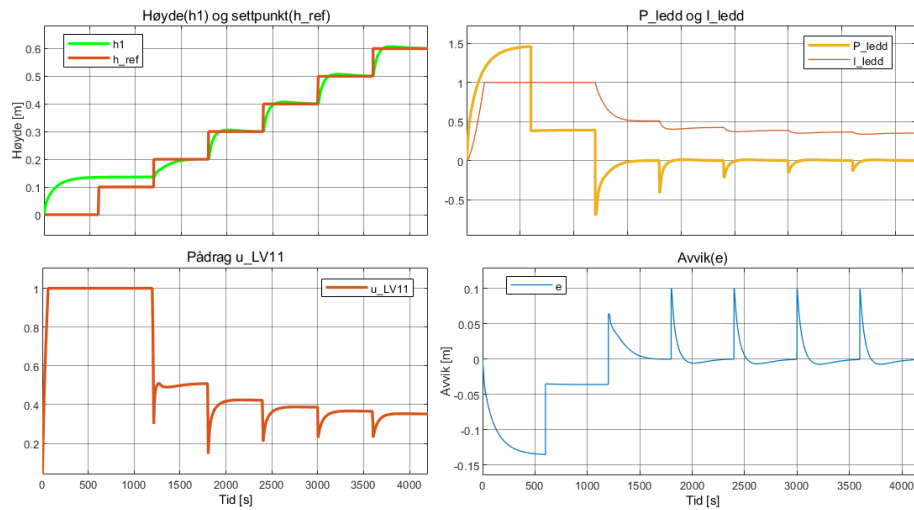
For å få til **Gain Scheduling** i Simulink ble vektorene for K_p og K_i implementert i regulatoren ved hjelp av **Lookup Tables** vist i figur 5.6. Ved å bruke **Lookup Tables** henter regulatoren ut riktige K_p og K_i verdier basert på høyden i tanken.



Figur 5.6: Figuren viser regulatoren til ventil LV11 med **Lookup Tables** markert i rød firkant. Regulatoren er konstruert med blokker fra **plclib** som er kompatibel med Simulink PLC Coder.

5.2 Nivåregulering tank 1: Ventil LV11

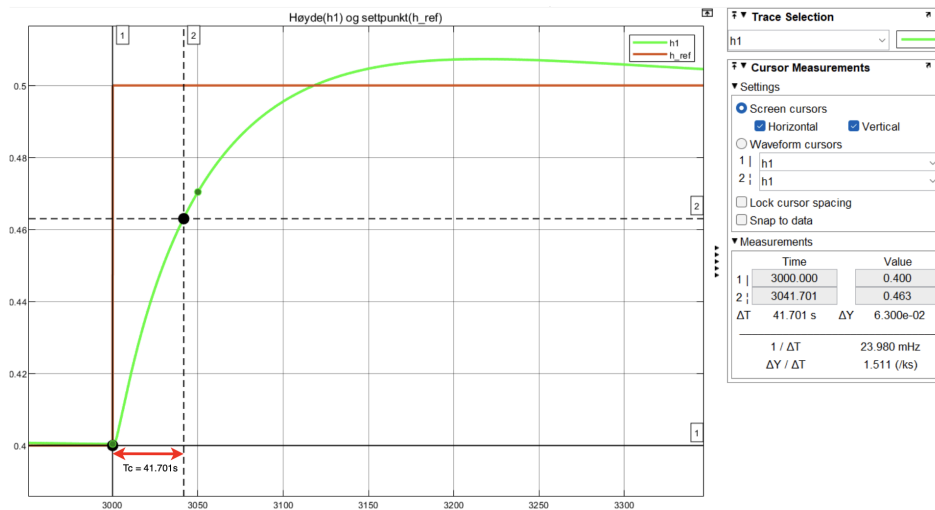
I figur 5.7 ser vi simuleringresultatet av nivåreguleringen av høyden i tanken. Høyden går ikke under 13.6cm grunnet trykkfall i tanken, derfor kan vi neglisjere de tre første påførte sprangene.



Figur 5.7: Figuren viser simuleringresultatet av høyden i tanken, P og I leddet til regulatoren, pådraget til ventil LV11 og avviket til høyden.

5.2 Nivåregulering tank 1: Ventil LV11

For å verifisere at vi får tilnærmet ønsket tidsrespons ($T_c = 50s$) ble denne målt ved å bruke verktøyet **cursors** i Simulink. Simuleringsresultatet viser at responsen har en initial respons tilsvarende et førsteordenssystem, men får oversving som et andreordenssystem. Oversvingen oppstår på grunn av nullpunkt i venstre halvdel av planet. Dette kommer fra den lukket sløyfe overføringsfunksjonen. Resultatet av den målte T_c verdien og dynamikken til systemet ved et påført sparang på 0.1m i arbeidspunktet er vist i figur 5.8.



Figur 5.8: Prosess for avlesning av T_c verdien. Avlest $T_c = 41.701$ s

Kapittel 6

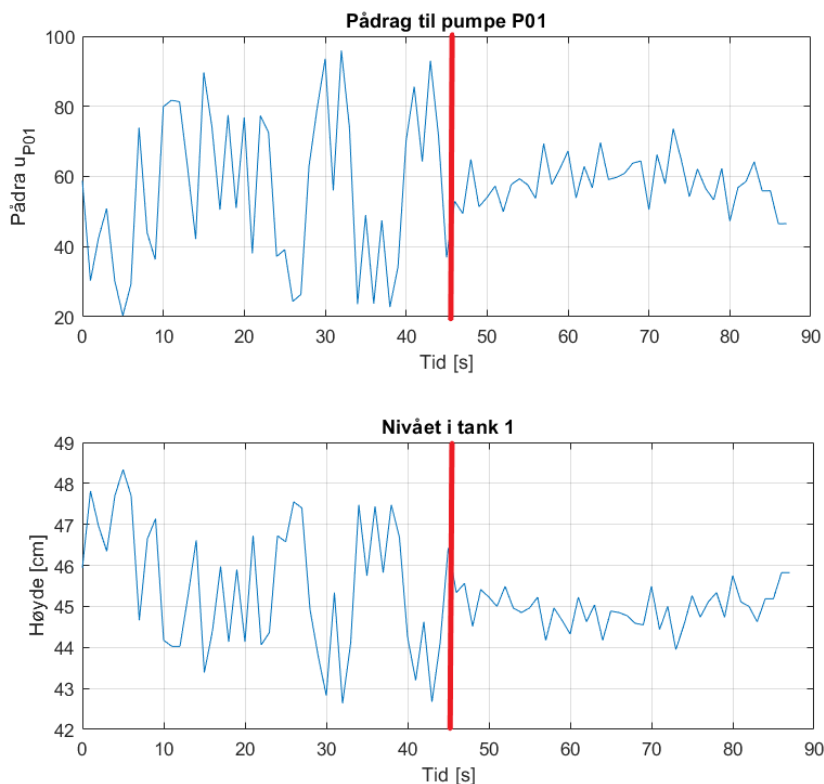
Regulering mot virkelig prosess

Vi skal i dette kapitlet se på sammenhengen mellom den overførte regulatoren fra Simulink PLC Coder, den innebygde regulatoren i Sysmac Studio og den teoretiske (simulerte) regulatoren fra Simulink.

6.1 Reguleringsproblematikk med overført pumpe-regulator

Når vi testkjørte den overførte pumpe regulatoren oppdaget vi store avvik på pumpepådraget som ikke var representativt med den teoretiske simuleringen. Under kjøring gikk pumpen ujevnt, hvor pådraget alternerte mellom 0% og 100%. Vi oppdaget store vibrasjoner ved kjøring av pumpe P01, prøvde derfor å redusere vibrasjonene ved å fysisk holde pumpen i ro. Dette reduserte alterneringen i pumpepådraget kraftig, for å vise effekten av å fysisk holde på pumpen se figur 6.1.

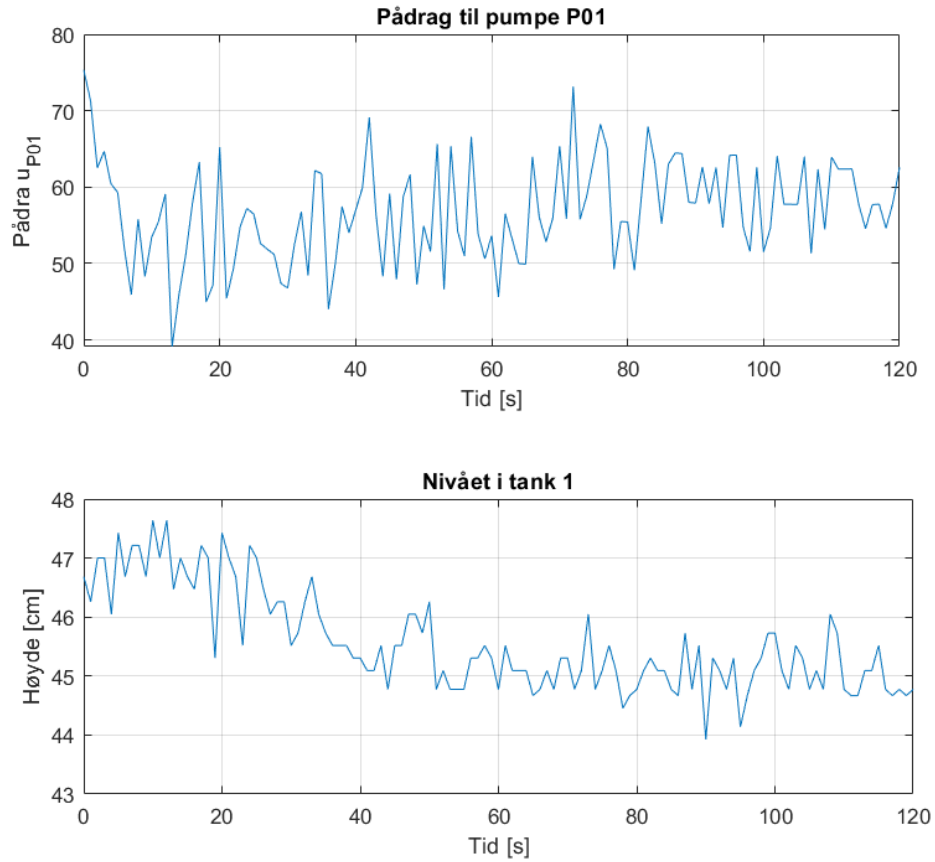
6.1 Reguleringsproblematikk med overført pumperegulator



Figur 6.1: Figuren viser forskjellen på pådraget til pumpe P01 og nivået i tank 1 når vi fysisk holder på pumpen. Når vi holder på pumpen reduseres vibrasjonene. Venstre side av figuren (til venstre for den røde streken) indikerer vanlig drift og høyre side av figuren (til høyre for den røde streken) viser pumpepådraget og nivået i tanken ved å fysisk holde pumpen i ro.

I forsøket på å redusere alterneringen i pumpepådraget forsøkte vi å koble ny IO-link nivåmåler opp mot tank 1. Vi ser i figur 6.2 at pådraget til pumpe P01 og nivået i tank 1 er representativt med målingene fra høyre siden av den røde streken i figur 6.1. Vi har konkludert med at vibrasjonene og pådragsalterneringen oppstår fra en forplantningsfeil i den eksisterende nivåmåleren. Et forbedringspotensial vil være å bytte den eksisterende nivåmåleren ut med en ny IO-link nivåmåler og konstruere et infinite impulse response(IIR)-filter.

6.1 Reguleringsproblematikk med overført pumperegulator



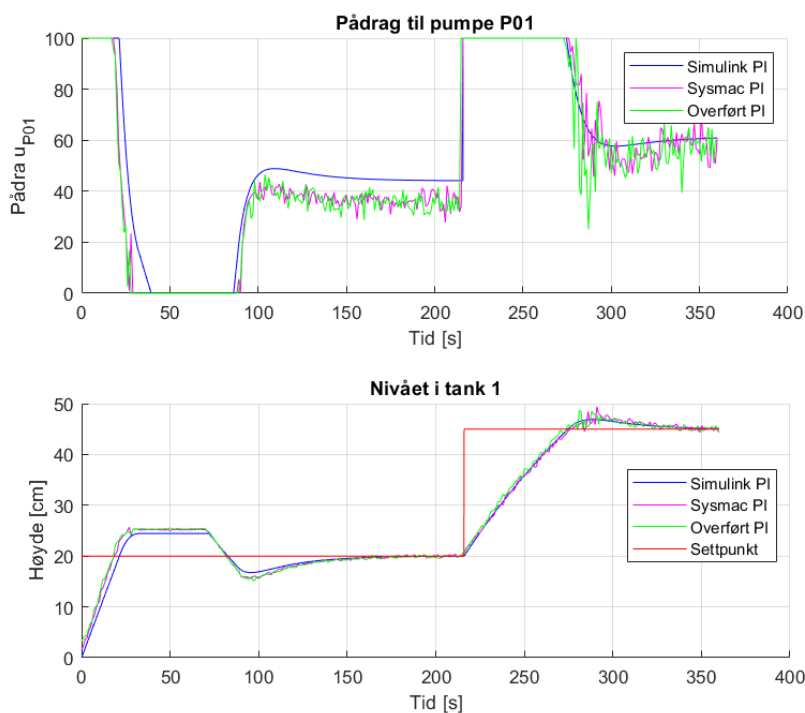
Figur 6.2: Figuren viser pådraget til pumpe P01 og nivået i tank 1 når vi kjørte reguleringen med IO-link nivåmåler.

6.2 Overført regulator: pumpe P01

6.2 Overført regulator: pumpe P01

Nivåregulering av tank 1 fra seksjon 5.1 ble implementert på den virkelige tretankprosessen ved å overføre generert ST kode som beskrevet i seksjon 4.1.1 og 4.1.2. Vi sammenliknet den overførte regulatoren med den innebygde PI regulatoren til Sysmac Studio og det simulerte resultatet fra Simulink. Regulator verdier er vist i likning 6.1 og resultatet av reguleringen er vist i figur 6.3.

$$\begin{aligned} K &= \frac{\Delta y}{\Delta u} = \frac{h_{max}/s}{1} = \frac{0.6}{64.15} \approx 0.00935 \\ T_C &= 7.2 \\ K_p &= 14.85 \\ T_i &= 28.8 \\ K_i &\approx 0.52 \end{aligned} \tag{6.1}$$

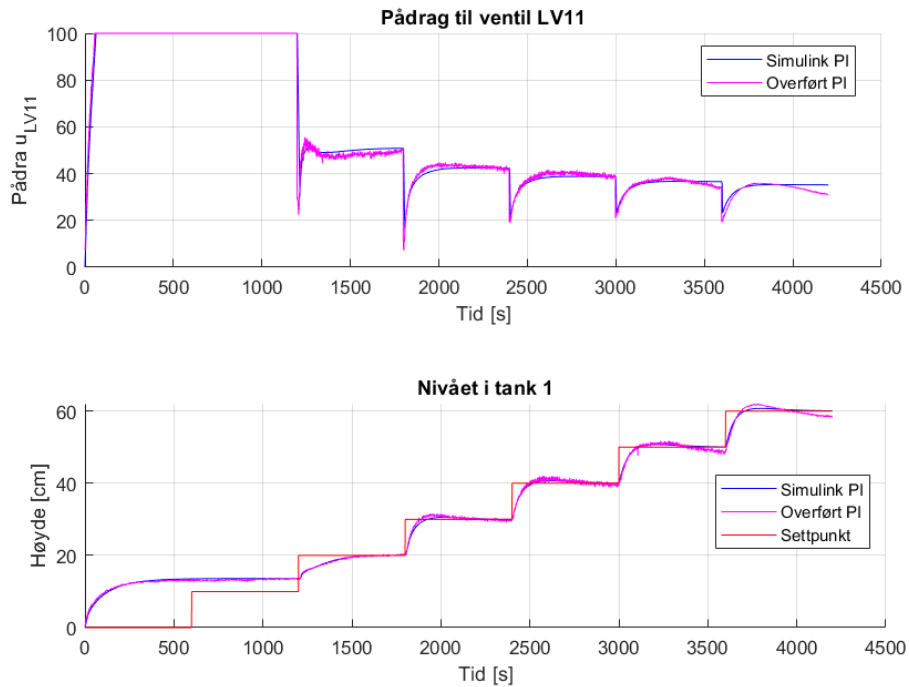


Figur 6.3: Figuren viser reguleringsresultatet av den overførte(PLC Coder), innebygde(Sysmac Studio) og Simulink PI regulatoren. Uteliggerene som oppsto i pumpepådraget ved ca. 285s og 340s skyldes at vi endret håndposisjon på pumpen.

6.3 Overført regulator: ventil LV11

6.3 Overført regulator: ventil LV11

Nivåregulering av tank 1 ved å anvende **Gain Scheduling** som beskrevet i seksjon 5.2 ble implementert på den virkelige tretankprosessen ved å overføre generert ST kode som beskrevet i seksjon 4.1.1 og 4.1.2. Vi sammenliknet den overførte regulatoren med det simulerte resultatet fra Simulink. K_p og K_i verdier som ble brukt ved de ulike nivåene er vist i plott 5.5 og resultatet av reguleringen er vist i figur 6.4.



Figur 6.4: Figuren viser reguleringsresultatet av den overførte PI regulatoren (PLC Coder) og Simulink PI regulatoren. Høyden går ikke under 13.6cm og følger derfor ikke settpunktet grunnet trykkfall i tanken, derfor kan vi neglisjere de tre første påførte sprangene.

Kapittel 7

Konklusjon

Vi oppgraderte tretankanlegget for å kunne bruke det med Sysmac Studio og Simulink PLC Coder, anlegget kan nå brukes i undervisning. Forbedringspotensial vil være å oppgradere til tre IO-Link nivååmalere. Vi programerte et nytt PLS og HMI program for å kunne styre systemet og se bruken av Sysmac Studio sammen med Simulink PLC Coder. Vi konstruert en matematisk modell som krevde at vi beregnet pumpe og ventilkarakterestikk. Den matematiske modellen ble brukt til å simulere det virkelige tretankanlegget offline. Reguleringsmetoder vi har sett på er nivå regulering av tank 1 ved å anvende PI regulator på en pumpe og **Gain Scheduling** på en ventil. Videre arbeid for å regulere tretankanlegget vil være å se på kaskaderegulering, bakover- og foroverkoboling. Vi brukte verktøyet Simulink PLC Coder for å generere ST fra PI regulatorene. Generering og overføring av ST kode fra Simulink til ønsket IDE var enkelt å gjennomføre ved å benytte oss av Simulink PLC Coder. Vi konkluderte med at Simulink PLC Coder er mer tidkrevende å bruke for enkle beregninger, men er et nyttig verktøy for mer avanserte beregninger. Fremtidig arbeid med Simulink PLC Coder kan være å undersøke flere bruksområder til verktøyet, foreksempel flowchart eller for andre bruksområder enn reguleringsteknikk.

Bibliografi

- [1] F. Haugen. Finn haugen: Pid control. http://techt teach.no/presentations/tekna_olje_gass_04/lecture/documents/skogestad.pdf, 2009.
- [2] MathWorks. Matlab test bench requirements and best practices for code generation. <https://se.mathworks.com/help/hdlcoder/ug/matlab-test-bench-best-practices.html>, åpnet: 06.05.2022.
- [3] MathWorks. Test bench verification. <https://se.mathworks.com/help/plccoder/ug/test-bench-verification.html>, åpnet: 06.05.2022.
- [4] MathWorks. Pdf documentation for simulink plc coder. https://se.mathworks.com/help/pdf_doc/plccoder/index.html, åpnet: 09.05.2022.
- [5] MathWorks. Supported blocks. <https://www.mathworks.com/help/plccoder/ug/supported-simulink-blocks.html>, åpnet: 20.04.2022.
- [6] Wikipedia. Fil:ventilkarakteristikk.gif. <https://nn.wikipedia.org/wiki/Fil:Ventilkarakteristikk.gif>, åpnet: 21.04.2022.

Vedlegg A

Programlisting

A.1 Overleaf

A.2 Matlab

A.3 Simulink

A.3.1 Simulink PLC Coder

A.4 Sysmac studio

Vedlegg B

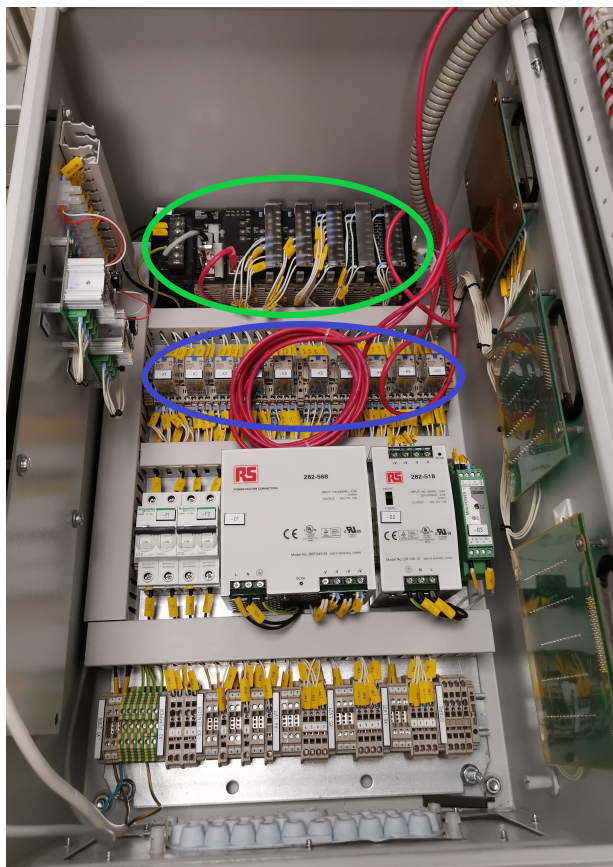
Ombygging

B.1 Ombygging av styreskap

B.1.1 Innmat

Utstyret som ble erstattet i det gamle skapet er avbildet i figur B.1. Releene ble fjernet og eksisterende PLS ble byttet ut med ny levert av Omron.

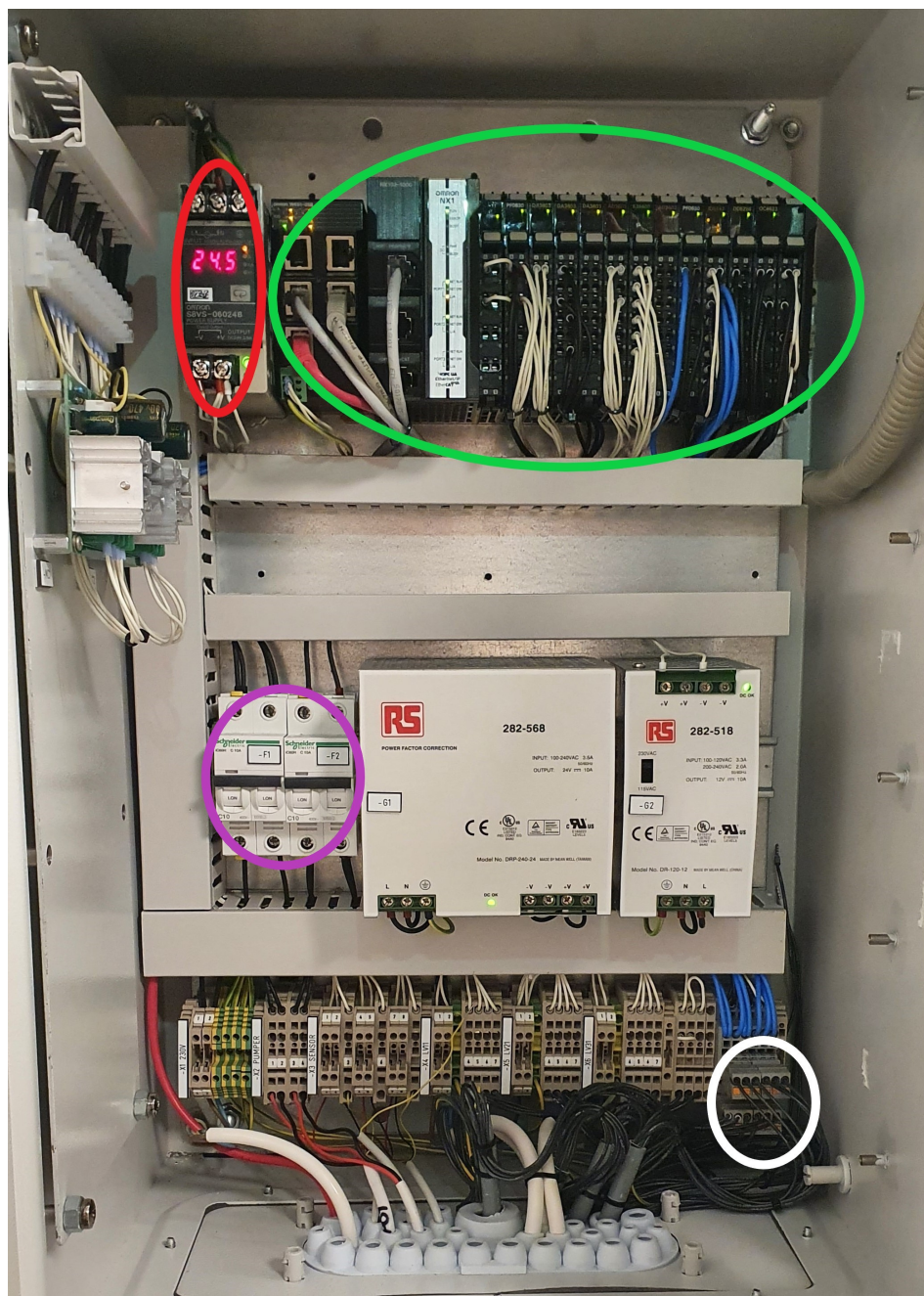
B.1 Ombygging av styreskap



Figur B.1: Styreskapet før ombygging. Grønn sirkel: PLS, blå sirkel: rele

Figur B.2 viser innmaten til skapet etter ombygging. Den nye PLS-en er montert øverst i skapet sammen med analoge -og digitale inngangs -og utgangskort. I tillegg monterte vi en ekstra 24 volts POWER SUPPLY som driver PLS og HMI. POWER SUPPLYEN har en egen sikring, F1, slik at hovedstrømmen og styrestrømmen blir separert. Dette er gjort for å øke anleggets driftssikkert, ved at eventuelle feil på pumpene og ventilene ikke skal påvirke PLS og HMI. Vi installerte 6 nye rekkeklemmer for å få nok tilkoblinger til endebryterene som er beskrevet i seksjon B.2.2.

B.1 Ombygging av styreskap

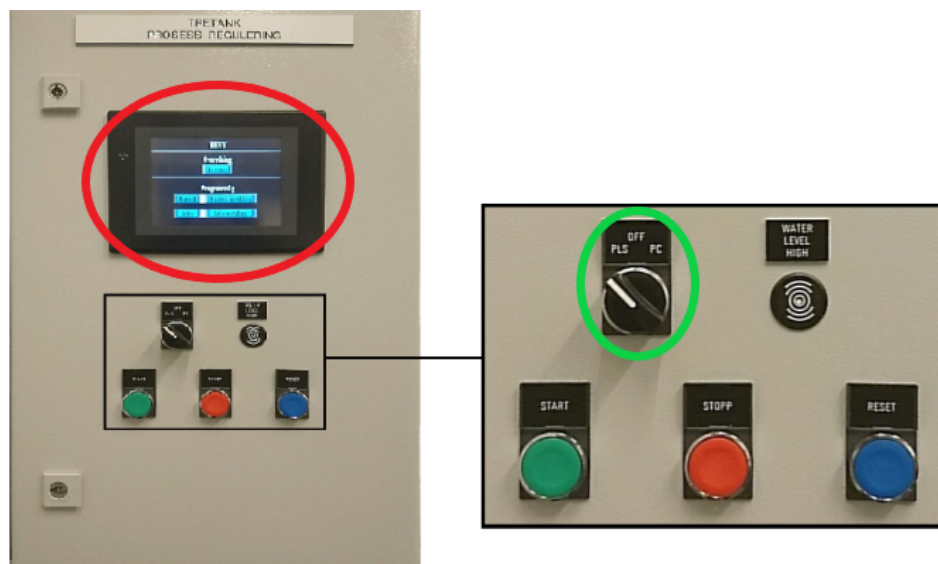


Figur B.2: Styreskapet etter ombygging. Rød sirkel: 24 volts power supply, grønn sirkel: PLS, lilla sirkel: sikring F1 og F2, hvit sirkel: ekstra rekkeklemme

B.1 Ombygging av styreskap

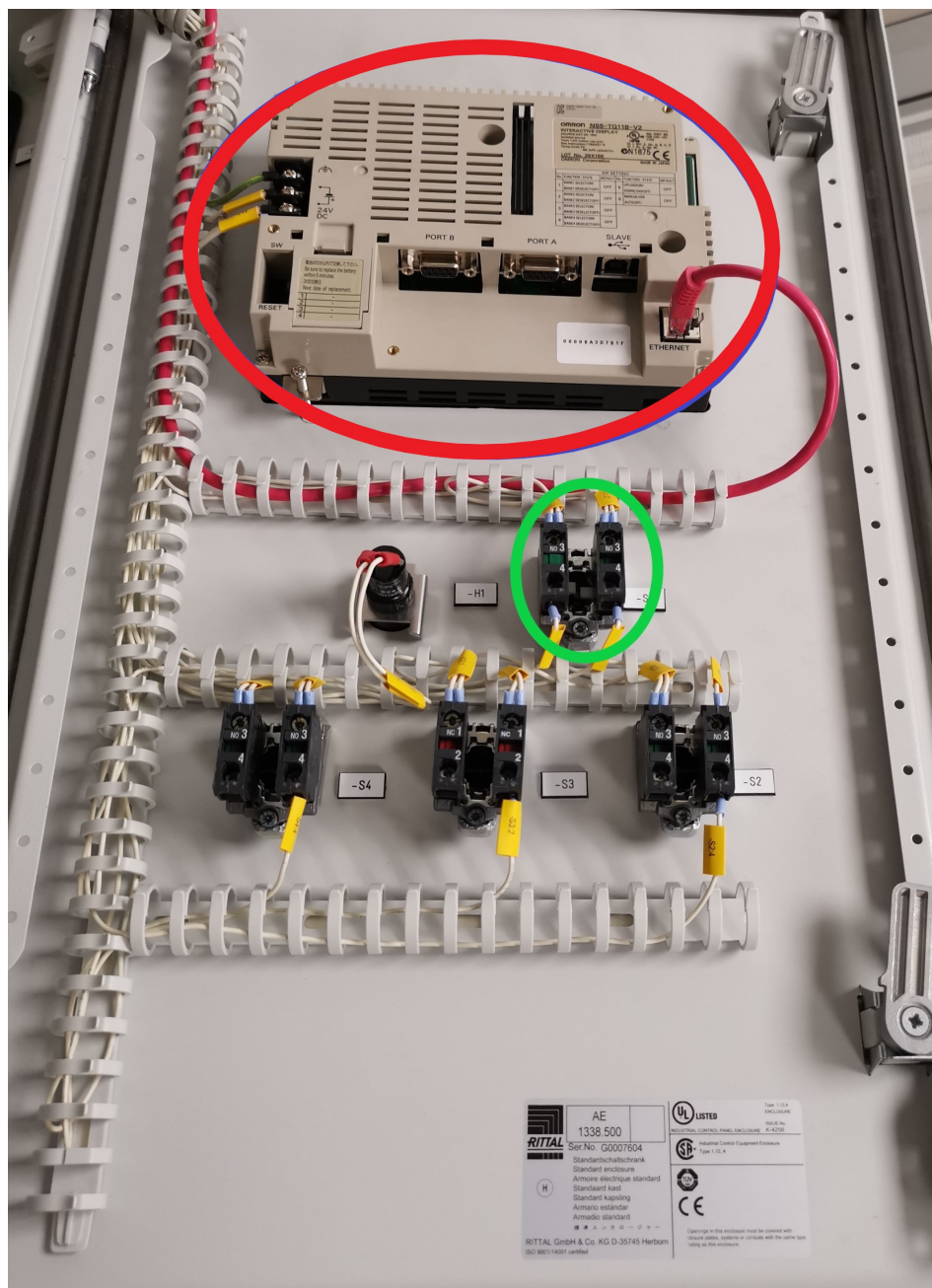
B.1.2 Skapdør

Figur B.3 og B.4 viser skapdør med gammel HMI og vribryter som ble erstattet.



Figur B.3: Fremside av skapdør før ombygging. Rød sirkel: HMI-panel, grønn sirkel: vribryter

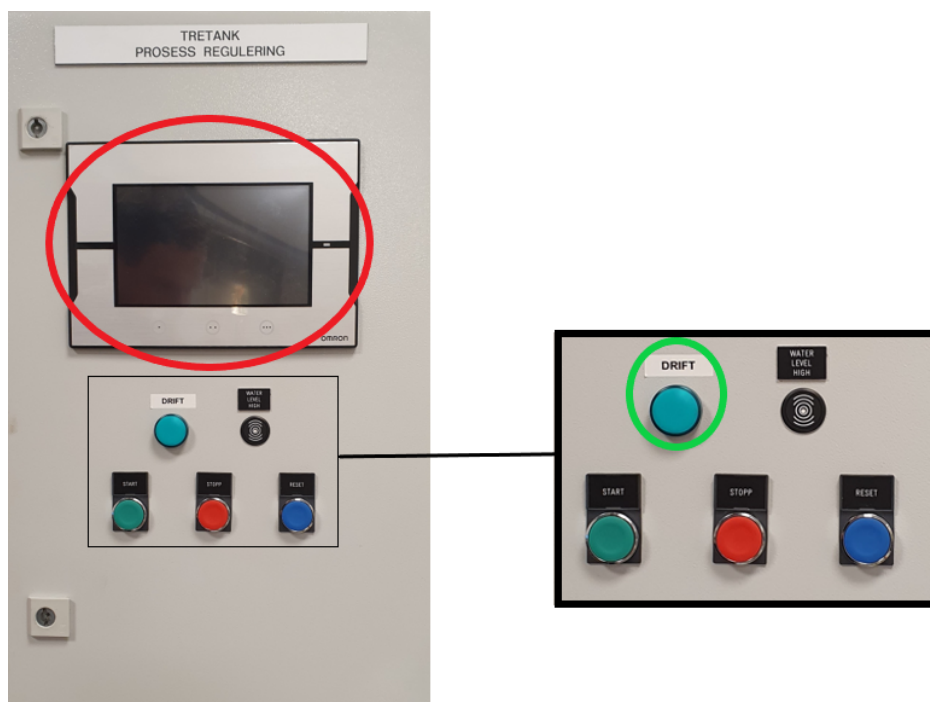
B.1 Ombygging av styreskap



Figur B.4: Bakside av skapdøren før ombygging. Rød sirkel: HMI, grønn sirkel: vibryter

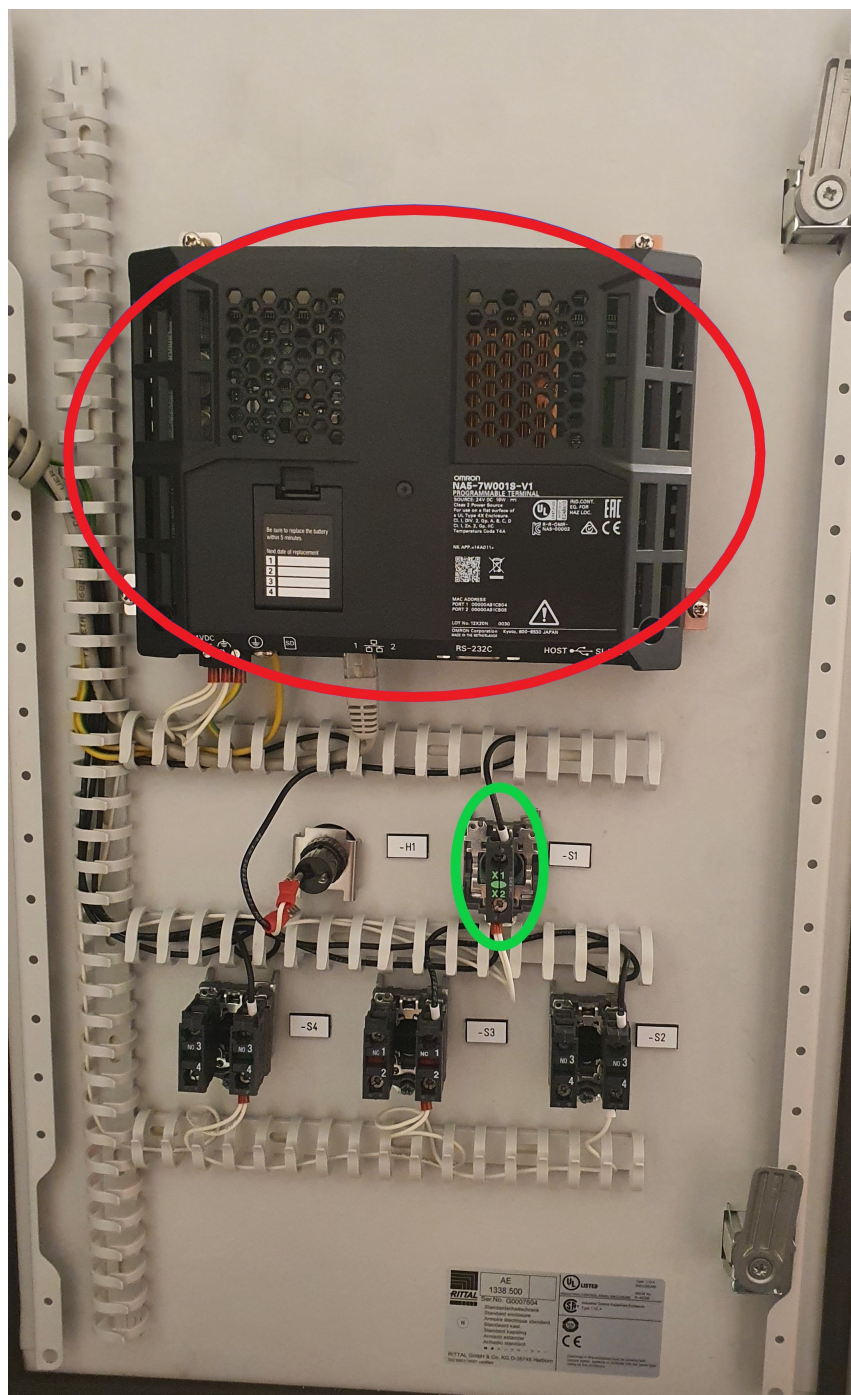
B.1 Ombygging av styreskap

Figur B.5 og B.6 viser skapdøren med ny HMI. For å montere det nye HMI panelet måtte hullet utvides, dette ble løst ved å levere HMI og skapdør til mekanisk verksted for kutting og montering. Vribryteren ble erstattet med en lampe som indikerer drift.



Figur B.5: Fremside av skapdør etter ombygging. Rød sirkel: HMI-panel, grønn sirkel: lampe.

B.1 Ombygging av styreskap



Figur B.6: Baksiden av skapdøren etter ombygging. Rød sirkel: HMI, grønn sirkel: lampe

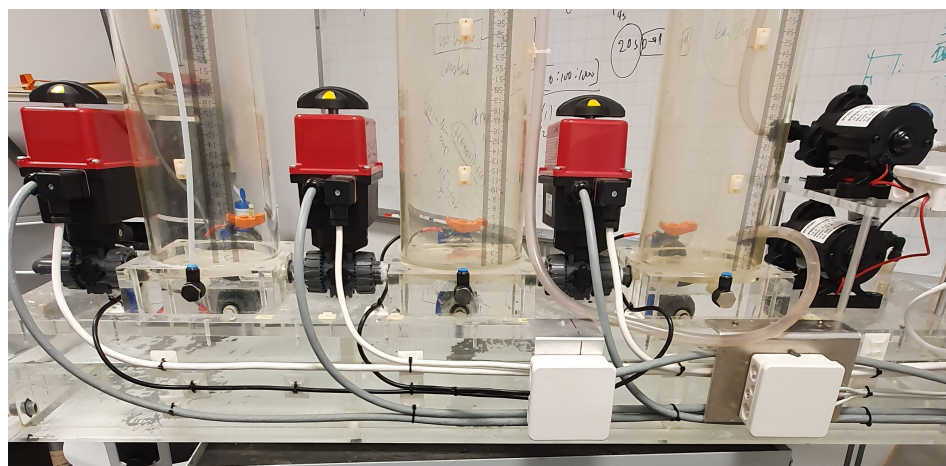
B.2 Tretank oppgradering

B.2 Tretank oppgradering

På grunn av defekte komponenter, dårlig utførte koblinger og mangel på ønsket funksjonalitet ble det gjort endringer og oppgraderinger på tretank anlegget.

B.2.1 Utbedring av eksisterende tretank anlegg

For å sikre mot kortslutning og strømgjennomgang ble koblingene utført i koblingsbokser av IP grad 65. Modellen ble utstyrt med stripsfester for å feste kablene så de ikke ligger løst. Resultatet er vist i figur B.7.



Figur B.7: Bilde av kobling på tretank.

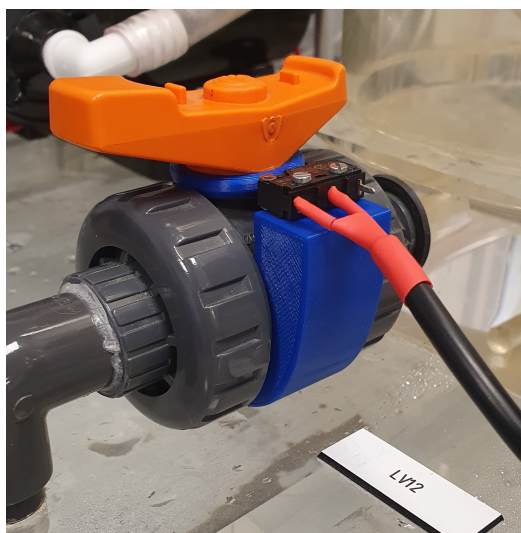
B.2.2 Montering av endebrytere til LV12, LV22 og LV32

Det ble montert endebrytere på ventilene LV12, LV22 og LV32 for å indikere om de er åpnet eller lukket, se figur B.8 og B.9.

B.2 Tretank oppgradering



Figur B.8: Endebyrter tilhørende manuell ventil med 3D printet trigger.

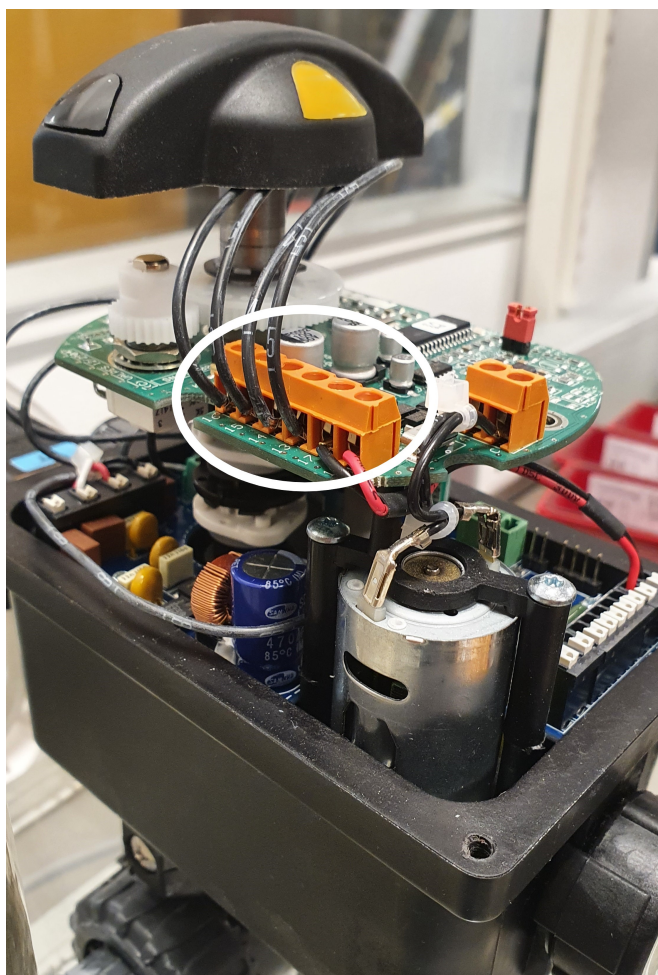


Figur B.9: Endebyrter tilhørende manuell ventil med 3D printet festeanordning.

B.2 Tretank oppgradering

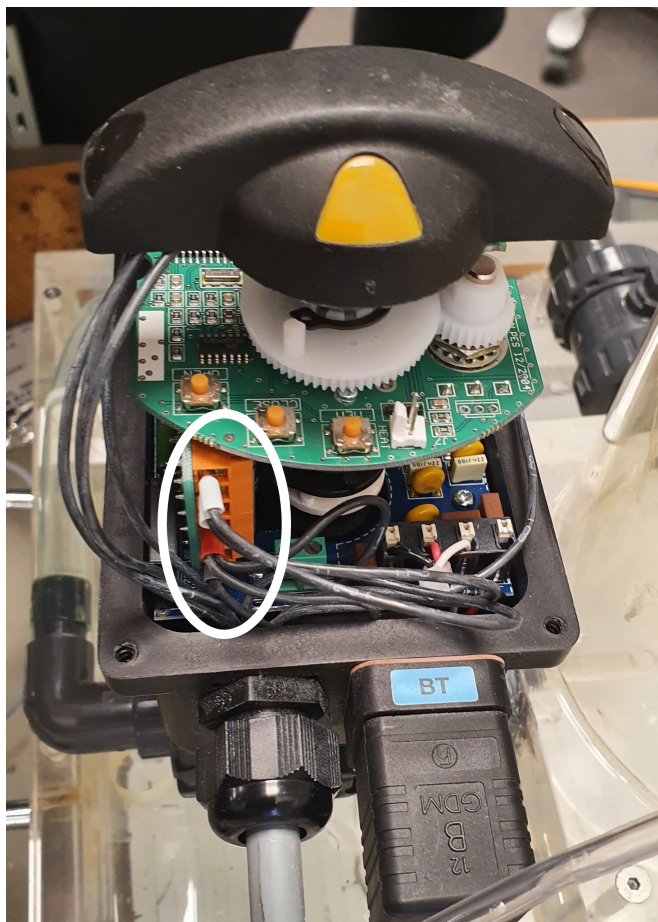
B.2.3 Omkobling av ventiler

I det gamle tretank anlegget var det ingen tilbakemelding fra de automatiske ventilene LV11, LV21 og LV31. For å få tilbakemelding måtte ventilene kobles om. Omkoblingen ble gjort ved å erstatte den eksisterende 4 leder kabelen til en kabel som inneholdt 8 ledere. Koblingen av disse ventilene er vist i figur B.10 og B.11. For mer detaljer om ventiloppkobling se vedlegg D og for mer info på ventilene se vedlegg C.2.1.



Figur B.10: Viser hvordan tilbakemeldingen på ventilene er koblet.

B.3 Feilsøking



Figur B.11: Viser hvordan tilførselen til ventilene er koblet.

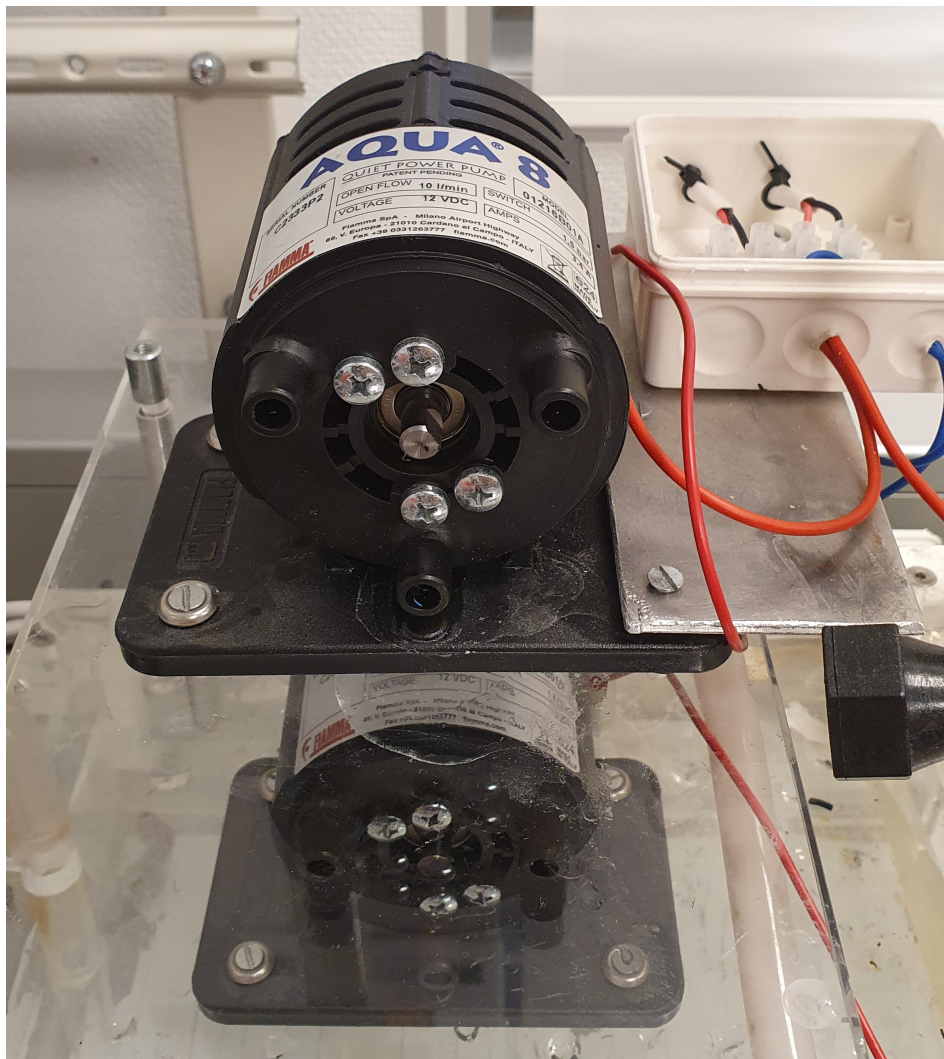
B.3 Feilsøking

B.3.1 Feilsøking på pumpene

Vi oppdaget at pumpe P01 laget u lyd og pumpe P02 ikke oppnådde ønsket turtall(se figur B.12 og B.13). For å finne årsaken til problemet ble begge pumpehusene demontert(se figur B.14). Det ble oppdaget defekte plastdeler i begge pumpehusene og defekt kulelager til akslingen i pumpe P02. Det ble

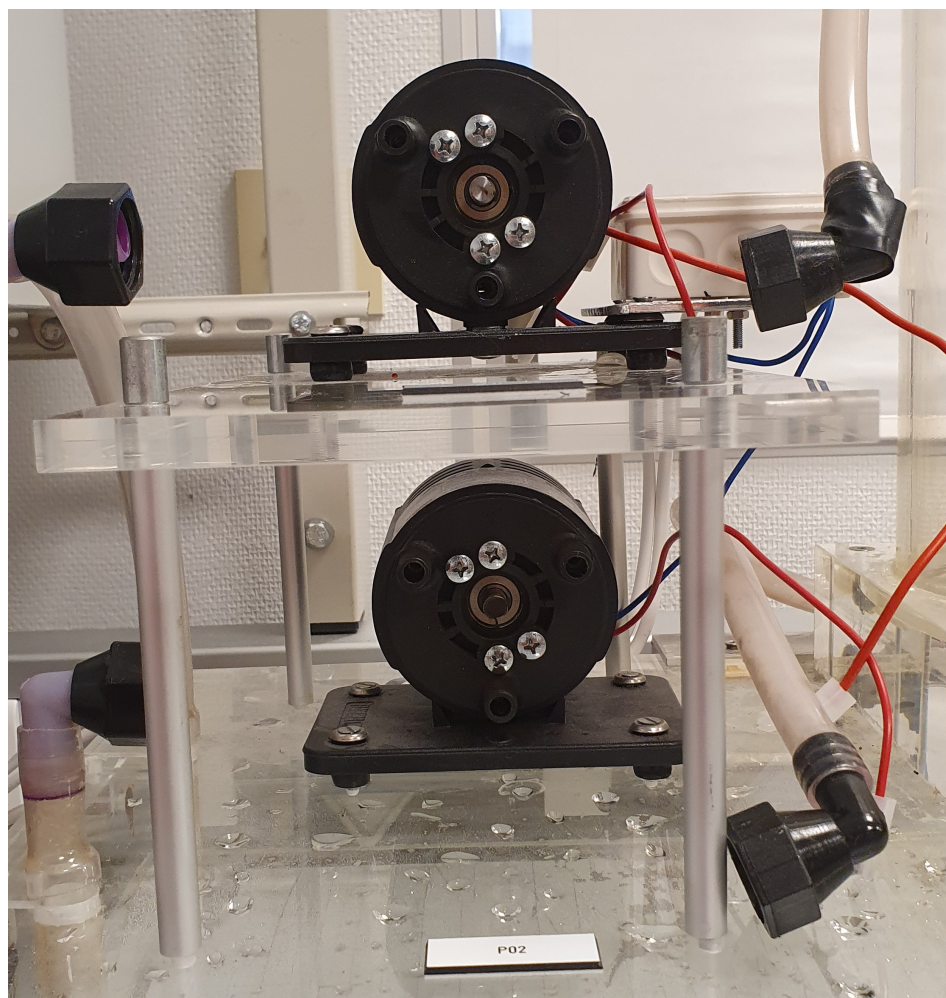
B.3 Feilsøking

derfor besluttet å kjøpe to nye pumper som var tilnærmet like de gamle. De nye pumpene er vist i figur B.15 og B.16.



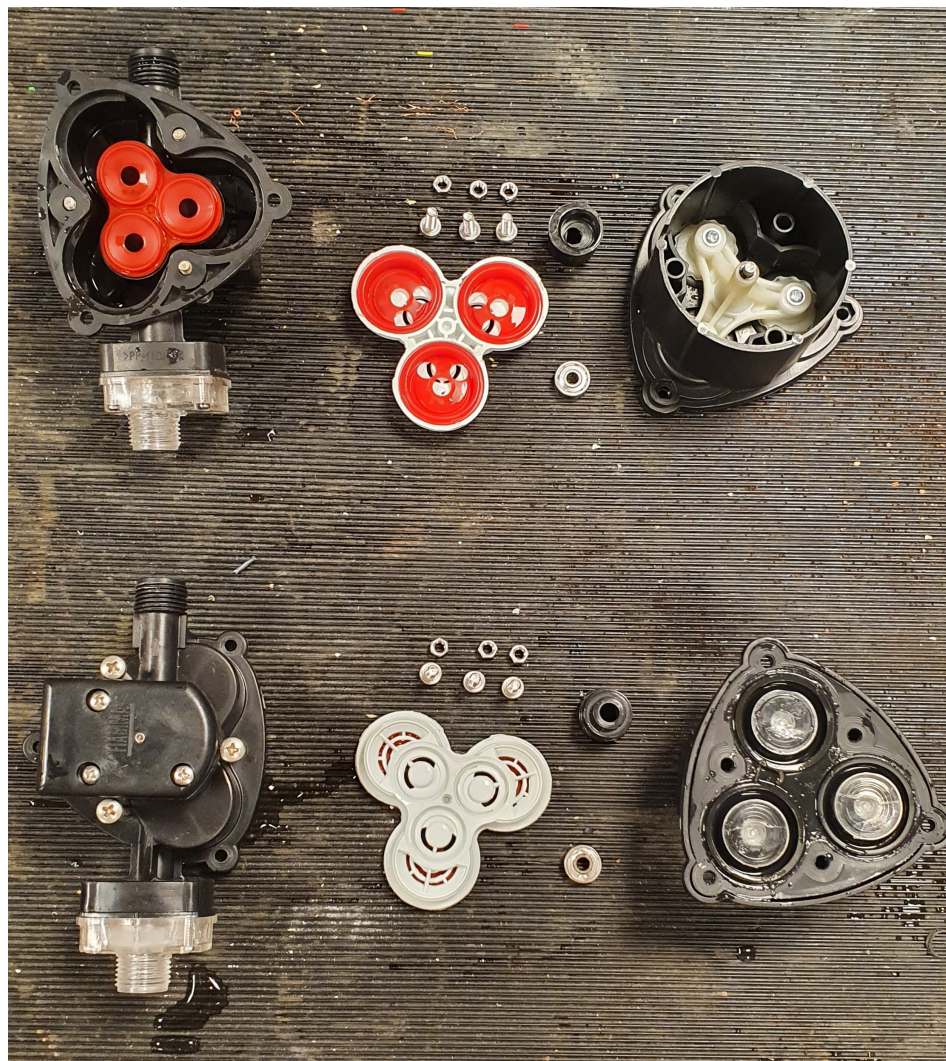
Figur B.12: Bilde av DC motor etter at pumpehuset er demontert.

B.3 Feilsøking



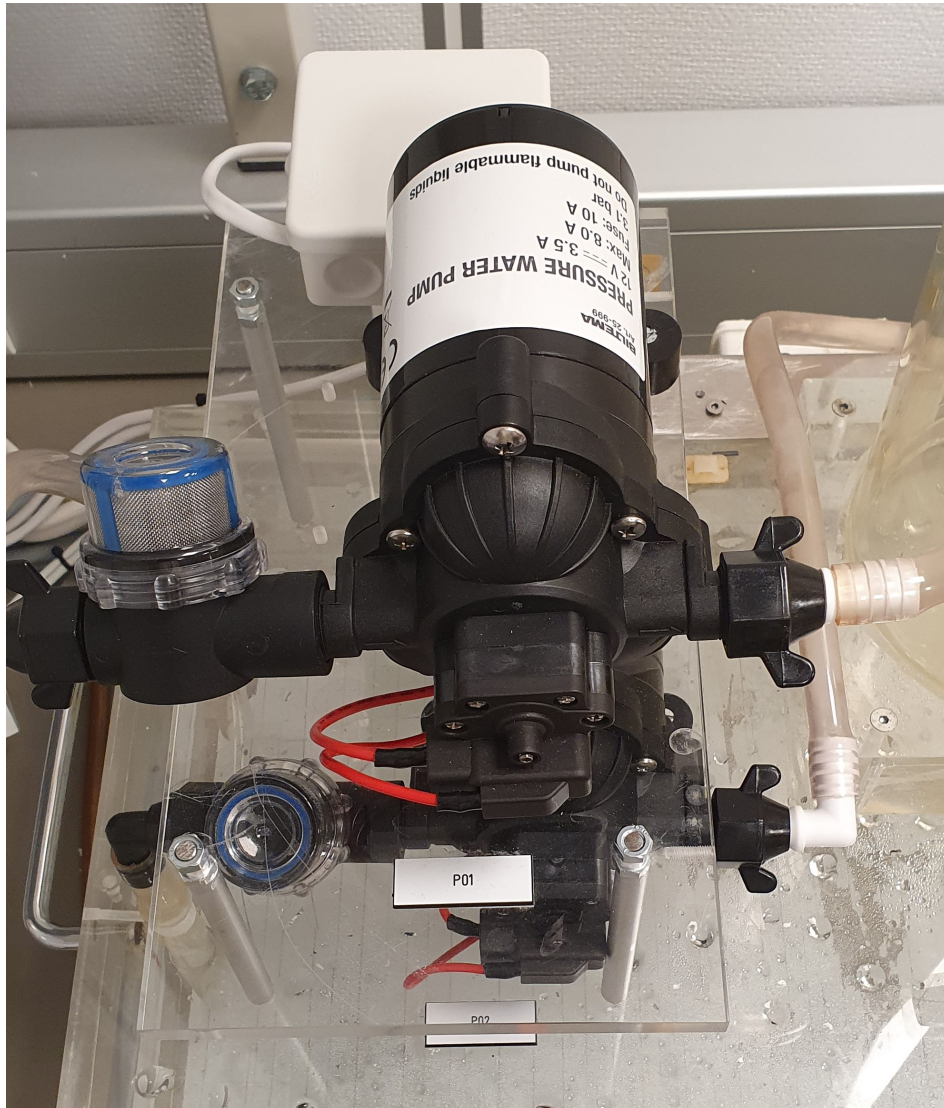
Figur B.13: Bilde av DC motorene etter at pumpehusene er demontert.

B.3 Feilsøking



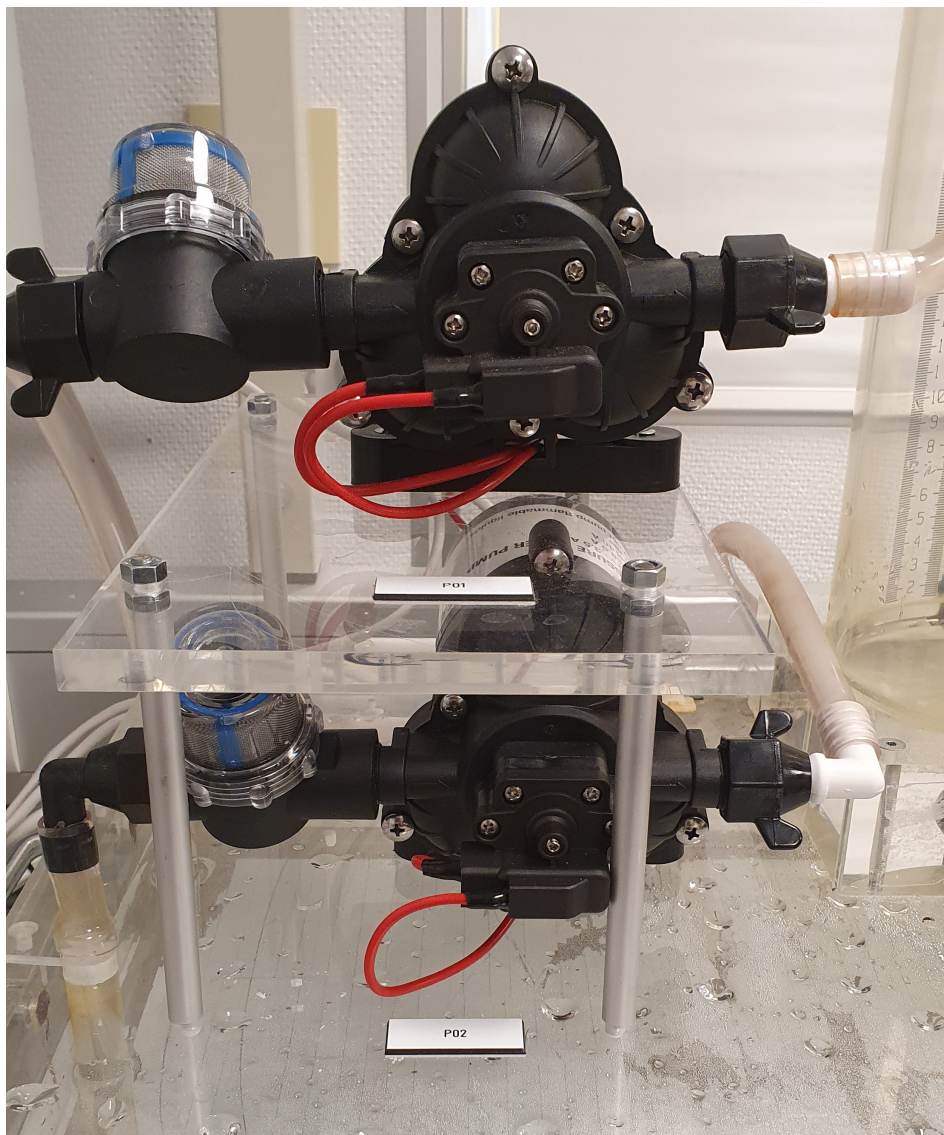
Figur B.14: Bilde av de defekte pumpehusene

B.3 Feilsøking



Figur B.15: Bilde av ny pumpe.

B.3 Feilsøking



Figur B.16: Bilde av ny pumpe.

B.3.2 Feilsøking på DP Celle

Vi oppdaget at DP celle, LT3, returnerte feil høyde når vi testet anlegget. Høyden som ble vist i HMI panelet stemte ikke overens med nivået som vi

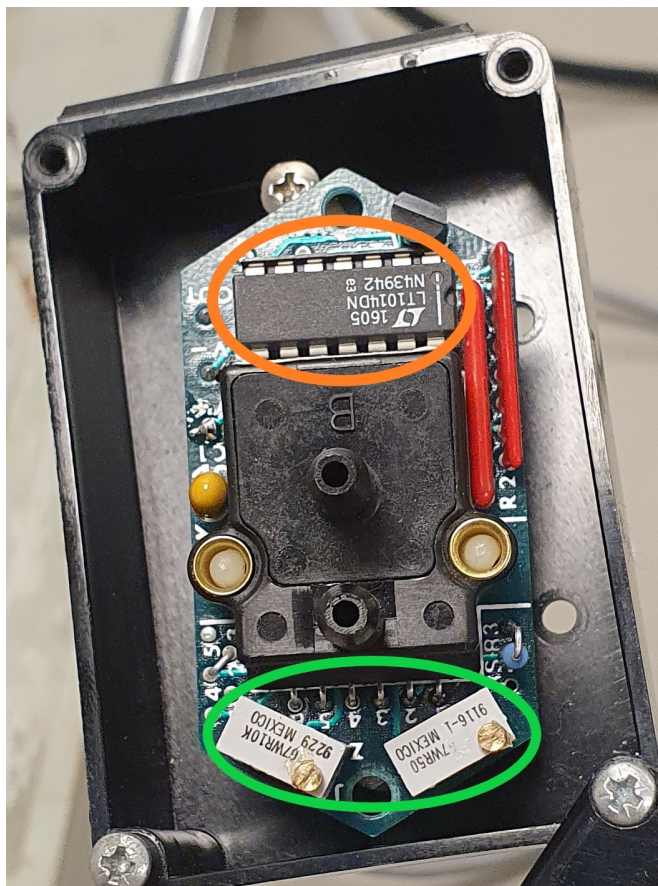
B.3 Feilsøking

leste av manuelt ved hjelp av målestokken i tank 3. Spenningen på kretskortet til DP cellen, LT3, hadde lavere verdi enn det databladet opplyste. I arbeidet med å finne årsaken til feilen prøvde vi både å justere på potensiometeret (kalibrering), skiftet ut komponenter (diode, IC), men feilen besto. Det ble derfor besluttet å kjøpe tre nye DP-celler. Figur B.17 og B.18 viser den defekte DP cellen.



Figur B.17: Bilde viser oversiden av kretskortet til DP cellen med diode (rød sirkel) tilkoblet mellom hvit leder og V_+ .

B.3 Feilsøking



Figur B.18: Bilde viser undersiden av kretskortet til DP cellen hvor man finner IC-en (oransj sirkel) og potensiometrene (grønn sirkel)

De nye nivåmålerene som ble bestilt bruker IO-link. Figur B.19 og B.20 viser hvordan disse er blitt montert på tretankanlegget. Vi monterte og koblet opp to stykker på grunn av lang leveringstid og forsinkelser på den tredje.

Vedlegg C

Datablad

C.1 Pumper

C.1.1 [Datablad til pumpe P01 og P02](#)

C.1.2 [Datablad til defekte pumper](#)

C.2 Ventiler

C.2.1 [Datablad til ventil LV11, LV21 og LV31](#)

C.3 Nivåmålere

C.3.1 [Datablad I/O-link nivåmåler](#)

C.3.2 [Datablad til nivåmåler LT1, LT2 og LT3](#)

C.4 Effektregulator

C.4 Effektregulator

C.4.1 [Datablad PWM Effektregulator](#)

C.5 Omron komponenter

C.5.1 [Datablad HMI](#)

C.5.2 [Datablad CPU](#)

C.5.3 [Datablad AI/AO Moduler](#)

C.5.4 [Datablad DI/DO Moduler](#)

C.5.5 [Datablad IO-link master modul](#)

C.5.6 [Datablad I/O Power Modul](#)

C.5.7 [Datablad Switch](#)

C.5.8 [Datablad IO-Link System](#)

C.5.9 [Datablad 24V powersupply](#)

C.5.10 [Omron CJ1](#)

C.6 Diverse Komponenter

C.6 Diverse Komponenter

C.6.1 [Datablad Sukkerbit](#)

C.6.2 [Datablad ABB AP-9](#)

C.6.3 [Datablad Rekkeklemme](#)

C.6.4 [Datablad Stripsfeste](#)

C.6.5 [Datablad Endebryter](#)

C.6.6 [Datablad Lampe](#)


Vedlegg D

Skjematiske tegninger

Trykk her for å åpne koblingstegningene for styreskapet til tretankanlegget 

Vedlegg E

Brukermanual

Trykk her for å åpne brukermanualen til tretankanlegget 


Vedlegg F

PLS program

F.1 Beskrivelse

Trykk her for å åpne beskrivelsen av PLS programmet 


F.2 Sysmac Studio kode

Trykk her for å laste ned selve PLS koden 


Trykk her for å åpne PLS koden som en PDF fil 


Vedlegg G

Matlab filer

Trykk her for Tretank fil med verdier. 

Trykk her for Tretank modell. 

Trykk her for Tretank modell med PI regulator til pumpe. 

Trykk her for Tretank modell med regulator til ventil. 

Vedlegg H

Kode

H.1 Kode for tretankanlegg verdier

```
1 clear all
2 close all
3
4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5 % Felles data
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7 rho = 997; % tetthet vann [kg/m^3]
8 g = 9.81; % tyngdens akselerasjon [m/s^2]
9 h_max = 0.6; % Maksimal h_Å,yde [m]
10 h_min = 0; % Minimal h_Å,yde [m]
11 A = 0.07^2 * pi; % Felles tverrsnittareal [m^2]
12 V = A * h_max; % Felles volum [m^3]
13 p = rho*g*h_max; % Trykket i full tank [Pa]
14
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16 % Pumpekarakteristikk P01
17 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
18 u_P01 = [0.17, 0.2, 0.4, 0.6, 0.8, 1];
19
20 q_P01 = [0, V/639.88, V/175.71, V/103.83, V/76.54, V/64.15];
21
22
23 figure
24 plot(u_P01, q_P01, '*-')
25 title('Pumpekarakteristikk')
26 xlabel('PÅÏdrag u_{P01}(t) til pumpe P01')
27 ylabel('VolumstrÅ,m q_{P01}(t) gjennom P01 [m^3/s]')
28
29 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
30 % Pumpekarakteristikk P02
31 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
32 u_P02 = [0.17, 0.2, 0.4, 0.6, 0.8, 1];
33
34 q_P02 = [0, V/603.76, V/171.36, V/105.61, V/77.72, V/65.19];
35
36
37 figure
38 plot(u_P02, q_P02, '*-')
```

H.1 Kode for tretankanlegg verdier

```
39 title('Pumpekarakteristikk')
40 xlabel('PÅÿdrag u_{P02}(t) til pumpe P02')
41 ylabel('VolumstrÅ,m q_{P02}(t) gjennom P02 [m^3/s]')
42
43 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
44 % Pumpekarakteristikk P03
45 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
46 u_P03 = [0, 0.05, 0.2, 0.4, 0.6, 0.8, 1];
47
48 q_P03 = [0, 0, V/249.01, V/120.42, V/82.65, V/65.81, V/61.85];
49
50
51
52 figure
53 plot(u_P03, q_P03, '*-')
54 title('Pumpekarakteristikk')
55 xlabel('PÅÿdrag u_{P03}(t) til pumpe P03')
56 ylabel('VolumstrÅ,m q_{P03}(t) gjennom P03 [m^3/s]')
57
58 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
59 % Fellesplott for pumpekarakteristikene
60 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
61
62 figure
63 hold on
64 plot(u_P01, q_P01, '*-')
65 plot(u_P02, q_P02, '*-')
66 plot(u_P03, q_P03, '*-')
67 title('Pumpekarakteristikk')
68 xlabel('PÅÿdrag u_{P0#}(t) til pumpene')
69 ylabel('VolumstrÅ,m q_{P0#}(t) gjennom pumpene [m^3/s]')
70 legend('P01', 'P02', 'P03')
71
72 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
73 % Ventilkarakteristikk LV11
74 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
75 u_LV11 = [0.25, 0.3, 0.4, 0.5, 0.6, 0.7];
76
77 q_LV11_maalt = [0, 2.208*10^(-5), 4.125*10^(-5), 5.989*10^(-5), ...
78 6.819*10^(-5), 7.458*10^(-5)];
79 q_LV11_max = q_LV11_maalt(1,6);
80
81 Kv_LV11 = q_LV11_max/sqrt(p); % ventilkonstant LV11 [m^3/(s*sqrt(Pa))]
82
83 f_LV11_beregnet = q_LV11_maalt/(Kv_LV11*sqrt(p));
84
85 figure
86 plot(u_LV11, f_LV11_beregnet, '*-')
87 title('Ventilkarakteristikk')
88 xlabel('PÅÿdrag u_{LV11}(t) til ventil LV11')
89 ylabel('f(u_{LV11}(t))')
90
91 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
92 % Ventilkarakteristikk LV21
93 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
94 u_LV21 = [0.25, 0.3, 0.4, 0.5, 0.6, 0.7];
95
96 q_LV21_maalt = [0, 2.189*10^(-5), 3.850*10^(-5), 5.732*10^(-5), ...
97 6.568*10^(-5), 7.069*10^(-5)];
98 q_LV21_max = q_LV21_maalt(1,6);
99
100 Kv_LV21 = q_LV21_max/sqrt(p); % ventilkonstant LV21 [m^3/(s*sqrt(Pa))]
101 f_LV21_beregnet = q_LV21_maalt/(Kv_LV21*sqrt(p));
102
103
104 figure
105 plot(u_LV21, f_LV21_beregnet, '*-')
106 title('Ventilkarakteristikk')
107 xlabel('PÅÿdrag u_{LV21}(t) til ventil LV21')
108 ylabel('f(u_{LV21}(t))')
109
110 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
111 % Ventilkarakteristikk LV31
112 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
113 u_LV31 = [0.25, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8];
114
```

H.1 Kode for tretankanlegg verdier

```
115 q_LV31_maalt = [0, 3.472*10^(-5), 6.689*10^(-5), ...
116 8.898*10^(-5), 11.016*10^(-5), 11.892*10^(-5), 12.323*10^(-5)];
117 q_LV31_max = q_LV31_maalt(1,7);
118
119 Kv_LV31 = q_LV31_max/sqrt(p); % ventilkonstant LV31 [m^3/(s*sqrt(Pa))]
120
121 f_LV31_beregnet = q_LV31_maalt/(Kv_LV31*sqrt(p));
122
123 figure
124 plot(u_LV31, f_LV31_beregnet, '*-')
125 title('Ventilkarakteristikk')
126 xlabel('PÅdrag u_{LV31}(t) til ventil LV31')
127 ylabel('f(u_{LV31}(t))')
128
129 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
130 % Fellesplott for ventilkarakteristikkene
131 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
132 figure
133 hold on
134 plot(u_LV11, f_LV11_beregnet, '*-')
135 plot(u_LV21, f_LV21_beregnet, '*-')
136 plot(u_LV31, f_LV31_beregnet, '*-')
137 title('Ventilkarakteristikk')
138 xlabel('PÅdrag u_{LV#1}(t) til ventilene')
139 ylabel('f(u_{LV#1}(t))')
140 legend('LV11', 'LV21', 'LV31')
141
142 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
143 % Ventilkarakteristikk Manuelle (LV12, LV22, LV32)
144 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
145 u_LV12 = [0, 1];
146 u_LV22 = [0, 1];
147 u_LV32 = [0, 1];
148
149 q_LV12_maalt = [0, 10.59*10^(-5)];
150 q_LV22_maalt = [0, 10.59*10^(-5)];
151 q_LV32_maalt = [0, 10.59*10^(-5)];
152 q_LV12_max = q_LV12_maalt(1,2);
153 q_LV22_max = q_LV22_maalt(1,2);
154 q_LV32_max = q_LV32_maalt(1,2);
155
156 Kv_LV12 = q_LV12_max/sqrt(p); % ventilkonstant LV12 [m^3/(s*sqrt(Pa))]
157 Kv_LV22 = q_LV22_max/sqrt(p); % ventilkonstant LV22 [m^3/(s*sqrt(Pa))]
158 Kv_LV32 = q_LV32_max/sqrt(p); % ventilkonstant LV32 [m^3/(s*sqrt(Pa))]
159
160 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
161 % PI verdier LV11
162 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
163 T = [135, 163, 217.5, 306, 393, 489.7];
164 Δ_h = [-0.025, -0.052, -0.093, -0.091, -0.105, -0.086];
165 Δ_u = [0.1, 0.1, 0.069, 0.0315, 0.0215, 0.012];
166
167 for i = 1:6
168 K(i) = Δ_h(i)/Δ_u(i);
169 end
170
171 Tc = 100;
172 for i = 1:6
173 Kp(i) = T(i)/(K(i)*Tc);
174 end
175 Kp = [Kp(1), Kp, Kp(6)];
176
177 for i = 1:6
178 Ki(i) = Kp(i)/min(T(i), 4*Tc);
179 end
180 Ki = [Ki(1), Ki, Ki(6)];
181
182 h_break = [0, 0.139, 0.164, 0.216, 0.309, 0.4, 0.505, 0.6];
```


H.2 Kode i Matlabfunction

H.2 Kode i Matlabfunction

```
1 function [hdot1, hdot2, hdot3] = fcn(h1, h2, h3, u_P01, u_P02, u_LV11, ...
   u_LV12, u_LV21, u_LV22, u_LV31, u_LV32)
2
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 % Div data
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6 rho = 997; % tetthet vann [kg/m^3]
7 g = 9.81; % tyngdens akselerasjon [m/s^2]
8 h_max = 0.6; % Maksimal hÅ,yde [m]
9 h_min = 0; % Minimal hÅ,yde [m]
10 A = 0.07^2 * pi; % Felles tverrsnittareal [m^2]
11 V = A * h_max; % Felles volum [m^3]
12 p = rho*g*h_max; % Trykket i full tank [Pa]
13
14 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15 % Trykk i tankene
16 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
17 p_LT1 = rho*g*h1;
18 p_LT2 = rho*g*h2;
19 p_LT3 = rho*g*h3;
20
21 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
22 % Ventilkonstanter [m^3/(s*sqrt(Pa))]
23 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
24 Kv_LV11 = 7.458 * 10^(-5) / sqrt(p);
25 Kv_LV12 = 10.59 * 10^(-5) / sqrt(p);
26 Kv_LV21 = 7.069 * 10^(-5) / sqrt(p);
27 Kv_LV22 = 10.59 * 10^(-5) / sqrt(p);
28 Kv_LV31 = 12.323 * 10^(-5) / sqrt(p);
29 Kv_LV32 = 10.59 * 10^(-5) / sqrt(p);
30
31 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
32 % Funksjon av ventil Åpning med hensyn pÅÿ pÅÿdrag.
33 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
34 if u_LV11 <= 0.25
35 f_LV11 = 0;
36 elseif u_LV11 >= 0.7
37 f_LV11 = 1;
38 else
39 f_LV11 = 8.583*u_LV11^3-16.93*u_LV11^2+12.02*u_LV11-2.062;
40 end
41
42 f_LV12 = u_LV12;
43
44 if u_LV21 <= 0.25
45 f_LV21 = 0;
46 elseif u_LV21 >= 0.7
47 f_LV21 = 1;
48 else
49 f_LV21 = 6.836*u_LV21^3-14.48*u_LV21^2+10.95*u_LV21-1.913;
50 end
51
52 f_LV22 = u_LV22;
53
54 if u_LV31 <= 0.25
55 f_LV31 = 0;
56 elseif u_LV31 >= 0.7
57 f_LV31 = 1;
58 else
59 f_LV31 = 4.774*u_LV31^3-11.14*u_LV31^2+9.163*u_LV31-1.641;
60 end
61
62 f_LV32 = u_LV32;
63
64 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
65 % VolumstrÅ_m igjennom pumper og ventiler
66 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
67
68 if u_P01 <= 0.17
69 q_P01 = 0;
```

H.3 Kode for verifisering

```
70 else
71     q_P01 = 6.826*10^(-6)*u_P01^3-8.924*10^(-5)*u_P01^2+...
72     0.266*10^(-3)*u_P01-3.933*10^(-5);
73 end
74
75 if u_P02 <= 0.17
76     q_P02 = 0;
77 else
78     q_P02 = 4.66*10^(-5)*u_P02^3-0.1611*10^(-3)*u_P02^2+...
79     0.2994*10^(-3)*u_P02-4.283*10^(-5);
80 end
81
82 q_LV11 = Kv_LV11*f_LV11*sign(p_LT1-p_LT2)*sqrt(abs(p_LT1-p_LT2));
83 q_LV12 = Kv_LV12*f_LV12*sqrt(p_LT1);
84 q_LV21 = Kv_LV21*f_LV21*sign(p_LT2-p_LT3)*sqrt(abs(p_LT2-p_LT3));
85 q_LV22 = Kv_LV22*f_LV22*sqrt(p_LT2);
86 q_LV31 = Kv_LV31*f_LV31*sqrt(abs(p_LT3));
87 q_LV32 = Kv_LV32*f_LV32*sqrt(abs(p_LT3));
88
89 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
90 % Deriverte hÅ,yde i tankene
91 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
92 hdot1 = 1/A*(q_P01 - q_LV11 - q_LV12);
93 hdot2 = 1/A*(q_P02 + q_LV11 - q_LV21 - q_LV22);
94 hdot3 = 1/A*(q_LV21 - q_LV31 - q_LV32);
```

H.3 Kode for verifisering

H.3.1 Verifisering tank 1

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % Verifisering tank 1
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 Tank1_data = importdata("MATLAB\bachelor\Logg Data Tank 1.csv");
5 Tank_I = Tank1_data.data([978:1704], 1);
6 Tid_Tank_1 = (0:0.5:363);
7 sim_data1 = out.sim_tank1{1}.Values.data;
8 sim_time1 = out.sim_tank1{1}.Values.time;
9 paadrag_P01 = out.sim_tank1{2}.Values.data;
10 paadrag_LV11 = out.sim_tank1{3}.Values.data;
11 paadrag_LV12 = out.sim_tank1{4}.Values.data;
12
13 figure
14 subplot(4,1,1)
15 plot(sim_time1, paadrag_P01)
16 title('PÅÿdrag til pumpe P01')
17 xlabel('Tid [s]')
18 ylabel('PÅÿdra u_{P01}')
19
20 subplot(4,1,2)
21 plot(sim_time1, paadrag_LV11)
22 title('PÅÿdrag til ventil LV11')
23 xlabel('Tid [s]')
24 ylabel('PÅÿdra u_{LV11}')
25
26 subplot(4,1,3)
27 plot(sim_time1, paadrag_LV12)
28 title('PÅÿdrag til ventil LV12')
29 xlabel('Tid [s]')
30 ylabel('PÅÿdra u_{LV12}')
31
32 subplot(4,1,4)
33 hold on
```

H.3 Kode for verifisering

```
34 plot(Tid_Tank_1, Tank_1)
35 plot(sim_time1, sim_data1*100)
36 title('Verifisering tank 1')
37 xlabel('Tid [s]')
38 ylabel('HÅyde [cm]')
39 legend('Virkelig','Simulert')
```

H.3.2 Verifisering tank 2

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % Verifisering tank 2
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 Tank2_data = importdata("TANK_2_HÅ^YDE.csv");
5 Tank_2 = Tank2_data.data([74:794], 1);
6 Tid_Tank_2 = (0:0.5:360);
7 sim_data2 = out.sim_tank2{1}.Values.data;
8 sim_time2 = out.sim_tank2{1}.Values.time;
9 paadrag_P01 = out.sim_tank2{2}.Values.data;
10 paadrag_P02 = out.sim_tank2{3}.Values.data;
11 paadrag_LV11 = out.sim_tank2{4}.Values.data;
12 paadrag_LV21 = out.sim_tank2{5}.Values.data;
13 paadrag_LV22 = out.sim_tank2{6}.Values.data;
14
15 figure
16 subplot(6,1,1)
17 plot(sim_time2, paadrag_P01)
18 title('PÅydrag til pumpe P01')
19 xlabel('Tid [s]')
20 ylabel('PÅydra u_{P01}')
21
22 subplot(6,1,2)
23 plot(sim_time2, paadrag_P02)
24 title('PÅydrag til pumpe P02')
25 xlabel('Tid [s]')
26 ylabel('PÅydra u_{P02}')
27
28 subplot(6,1,3)
29 plot(sim_time2, paadrag_LV11)
30 title('PÅydrag til ventil LV11')
31 xlabel('Tid [s]')
32 ylabel('PÅydra u_{LV11}')
33
34 subplot(6,1,4)
35 plot(sim_time2, paadrag_LV21)
36 title('PÅydrag til ventil LV21')
37 xlabel('Tid [s]')
38 ylabel('PÅydra u_{LV21}')
39
40 subplot(6,1,5)
41 plot(sim_time2, paadrag_LV22)
42 title('PÅydrag til ventil LV22')
43 xlabel('Tid [s]')
44 ylabel('PÅydra u_{LV22}')
45
46 subplot(6,1,6)
47 hold on
48 plot(Tid_Tank_2, Tank_2)
49 plot(sim_time2, sim_data2*100)
50 title('Verifisering tank 2')
51 xlabel('Tid [s]')
52 ylabel('HÅyde [cm]')
53 legend('Virkelig','Simulert')
```

H.4 Generert kode for pumpekontroller

H.3.3 Verifisering tank 3

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Verifisering tank 3
3  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4  Tank3_data = importdata("TANK_3_HÅYDE.csv");
5
6  Tank_3 = Tank3_data.data([43:745], 1);
7  Tid_Tank_3 = (0:0.5:351);
8  sim_data3 = out.sim_tank3{1}.Values.data;
9  sim_time3 = out.sim_tank3{1}.Values.time;
10 paadrag_P02 = out.sim_tank3{2}.Values.data;
11 paadrag_LV21 = out.sim_tank3{3}.Values.data;
12 paadrag_LV31 = out.sim_tank3{4}.Values.data;
13 paadrag_LV32 = out.sim_tank3{5}.Values.data;
14
15 figure
16 subplot(5,1,1)
17 plot(sim_time3, paadrag_P02)
18 title('PÅdrag til pumpe P02')
19 xlabel('Tid [s]')
20 ylabel('PÅdra u_{P02}')
21
22 subplot(5,1,2)
23 plot(sim_time3, paadrag_LV21)
24 title('PÅdrag til ventil LV21')
25 xlabel('Tid [s]')
26 ylabel('PÅdra u_{LV21}')
27
28 subplot(5,1,3)
29 plot(sim_time3, paadrag_LV31)
30 title('PÅdrag til ventil LV31')
31 xlabel('Tid [s]')
32 ylabel('PÅdra u_{LV31}')
33
34 subplot(5,1,4)
35 plot(sim_time3, paadrag_LV32)
36 title('PÅdrag til ventil LV32')
37 xlabel('Tid [s]')
38 ylabel('PÅdra u_{LV32}')
39
40 subplot(5,1,5)
41 hold on
42 plot(Tid_Tank_3, Tank_3)
43 plot(sim_time3, sim_data3*100)
44 title('Verifisering tank 3')
45 xlabel('Tid [s]')
46 ylabel('HÅyde [cm]')
47 legend('Virkelig', 'Simulert')
```

H.4 Generert kode for pumpekontroller

```
1  (*
2  *
3  * File: Tretank_Rev02.st
4  *
5  * IEC 61131-3 Structured Text (ST) code generated for subsystem ...
6  *   "Tretank_Rev02/Subsystem"
7  *
8  * Model name           : Tretank_Rev02
9  * Model version        : 2.4
10 * Model creator         : sondresalvesen
```

H.4 Generert kode for pumpekontroller

```
10 * Model last modified by      : sindr
11 * Model last modified on     : Wed May 11 15:47:44 2022
12 * Model sample time         : 0.002s
13 * Subsystem name            : Tretank_Rev02/Subsystem
14 * Subsystem sample time     : 0.002s
15 * Simulink PLC Coder version : 3.6 (R2022a) 13-Nov-2021
16 * Target Plugin version     : 2.20
17 * ST code generated on      : Thu May 12 18:00:45 2022
18 *
19 * Target IDE selection       : OMRON Sysmac Studio
20 * Test Bench included       : No
21 *
22 *)
23 FUNCTION_BLOCK Subsystem
24 VAR_INPUT
25     ssMethodType: SINT;
26     Kp: LREAL;
27     SP: LREAL;
28     Ki: LREAL;
29     PV: LREAL;
30 END_VAR
31 VAR_OUTPUT
32     MV: LREAL;
33     e: LREAL;
34     b_P: LREAL;
35     I: LREAL;
36 END_VAR
37 VAR
38     c_DiscreteTimeIntegrator_DS: LREAL;
39 END_VAR
40 CASE ssMethodType OF
41     SINT#0:
42         (* SystemInitialize for Atomic SubSystem: '<Root>/Subsystem' *)
43         (* InitializeConditions for DiscreteIntegrator: ...
44         '<S1>/Discrete-Time Integrator' *)
45         c_DiscreteTimeIntegrator_DS := LREAL#0.0;
46         (* End of SystemInitialize for SubSystem: '<Root>/Subsystem' *)
47     SINT#1:
48         (* Outputs for Atomic SubSystem: '<Root>/Subsystem' *)
49         (* Output: '<Root>/I' incorporates:
50         * DiscreteIntegrator: '<S1>/Discrete-Time Integrator' *)
51         I := c_DiscreteTimeIntegrator_DS;
52         (* Sum: '<S1>/Sum1' *)
53         e := SP - PV;
54         (* Product: '<S1>/Product2' *)
55         b_P := Kp * e;
56         (* Sum: '<S1>/Sum' incorporates:
57         * DiscreteIntegrator: '<S1>/Discrete-Time Integrator' *)
58         MV := b_P + c_DiscreteTimeIntegrator_DS;
59         (* Saturate: '<S1>/Saturation' *)
60         IF MV > LREAL#1.0 THEN
61             (* Output: '<Root>/MV' *)
62             MV := LREAL#1.0;
63         ELSIF MV < LREAL#0.0 THEN
64             (* Output: '<Root>/MV' *)
65             MV := LREAL#0.0;
66         END_IF;
67         (* End of Saturate: '<S1>/Saturation' *)
68
69         (* Update for DiscreteIntegrator: '<S1>/Discrete-Time ...
70         Integrator' incorporates:
71         * Product: '<S1>/Product1' *)
72         c_DiscreteTimeIntegrator_DS := ((e * Ki) * LREAL#0.002) + ...
73         c_DiscreteTimeIntegrator_DS;
74         IF c_DiscreteTimeIntegrator_DS > LREAL#1.0 THEN
75             c_DiscreteTimeIntegrator_DS := LREAL#1.0;
76         ELSIF c_DiscreteTimeIntegrator_DS < LREAL#-1.0 THEN
77             c_DiscreteTimeIntegrator_DS := LREAL#-1.0;
78         END_IF;
79         (* End of Update for DiscreteIntegrator: '<S1>/Discrete-Time ...
80         Integrator' *)
81     (* End of Outputs for SubSystem: '<Root>/Subsystem' *)
82 END_CASE;
83 END_FUNCTION_BLOCK
```

H.5 Generert kode for ventilkontroller

H.5 Generert kode for ventilkontroller

```
1  (*
2  *
3  * File: Tretank_LV11_PID.st
4  *
5  * IEC 61131-3 Structured Text (ST) code generated for subsystem ...
6  * "Tretank_LV11_PID/Subsystem"
7  *
8  * Model name           : Tretank_LV11_PID
9  * Model version        : 2.7
10 * Model creator         : sondresalvesen
11 * Model last modified by : andre.rabba
12 * Model last modified on : Thu Apr 21 18:27:57 2022
13 * Model sample time    : 0.002s
14 * Subsystem name       : Tretank_LV11_PID/Subsystem
15 * Subsystem sample time : 1s
16 * Simulink PLC Coder version : 3.6 (R2022a) 13-Nov-2021
17 * Target Plugin version  : 2.20
18 * ST code generated on  : Mon Apr 25 15:54:25 2022
19 *
20 * Target IDE selection   : OMRON Sysmac Studio
21 * Test Bench included    : No
22 *
23 *)
24 FUNCTION BLOCK Subsystem
25 VAR_INPUT
26   ssMethodType: SINT;
27   SP: LREAL;
28   PV: LREAL;
29 END_VAR
30 VAR_OUTPUT
31   MV: LREAL;
32 END_VAR
33 VAR
34   c_DiscreteTimeIntegrator_DS: LREAL;
35   LookupTablenD2_bp01Data: ARRAY [0..7] OF LREAL := ...
36     LREAL#0.0 ,LREAL#0.139 ,LREAL#0.164 ,LREAL#0.216 , ...
37     LREAL#0.309 ,
38     LREAL#0.4 ,LREAL#0.505 ,LREAL#0.6 ;
39   LookupTablenD3_tableData: ARRAY [0..7] OF LREAL := ...
40     LREAL#-8.0E-5 ,LREAL#-8.0E-5 , ...
41     LREAL#-3.8461538461538463E-5 ,
42     LREAL#-1.4838709677419359E-5 , ...
43     LREAL#-6.9230769230769251E-6 , ...
44     LREAL#-4.0952380952380959E-6 ,
45     LREAL#-3.4165116279069772E-6 , ...
46     LREAL#-3.4165116279069772E-6 ;
47   LookupTablenD2_tableData: ARRAY [0..7] OF LREAL := ...
48     LREAL#-18.518518518518519 ,LREAL#-18.518518518518519 ,
49     LREAL#-31.901840490797543 ,LREAL#-61.96901549225386 , ...
50     LREAL#-94.408133623819879 ,LREAL#-124.26770814841115 ,
51     LREAL#-146.3481042815329 ,LREAL#-146.3481042815329 ;
52   LookupTablenD3_bp01Data: ARRAY [0..7] OF LREAL := ...
53     LREAL#0.0 ,LREAL#0.139 ,LREAL#0.164 ,LREAL#0.216 , ...
54     LREAL#0.309 ,
55     LREAL#0.4 ,LREAL#0.505 ,LREAL#0.6 ;
56 END_VAR
57 VAR_TEMP
58   iRight: UDINT;
59   iLeft: UDINT;
60   bpIdx: UDINT;
61   iLeft_0: UDINT;
62   rtb_Product1: LREAL;
63   frac_tmp: LREAL;
64   temp1: DWORD;
65   temp2: DWORD;
66 END_VAR
67 CASE ssMethodType OF
68   SINT#0:
69     (* SystemInitialize for Atomic SubSystem: '<Root>/Subsystem' *)
```

H.5 Generert kode for ventilkontroller

```

65      (* InitializeConditions for DiscreteIntegrator: ...
66          '<S1>/Discrete-Time Integrator' *)
67      c_DiscreteTimeIntegrator_DS := LREAL#0.353;
68      (* End of SystemInitialize for SubSystem: '<Root>/Subsystem' *)
69      SINT#1:
70      (* Outputs for Atomic SubSystem: '<Root>/Subsystem' *)
71      (* Lookup_n-D: '<S1>/Lookup Table (n-D)2' incorporates:
72          * Lookup_n-D: '<S1>/Lookup Table (n-D)3' *)
73      (* Column-major Lookup 1-D
74          Search method: 'binary'
75          Use previous index: 'off'
76          Interpolation method: 'Linear point-slope'
77          Extrapolation method: 'Linear'
78          Use last breakpoint for index at or above upper limit: 'off'
79          Remove protection against out-of-range input in generated ...
80              code: 'off' *)
81      (* Prelookup - Index and Fraction
82          Index Search method: 'binary'
83          Extrapolation method: 'Linear'
84          Use previous index: 'off'
85          Use last breakpoint for index at or above upper limit: 'off'
86          Remove protection against out-of-range input in generated ...
87              code: 'off' *)
88      IF PV ≤ LREAL#0.0 THEN
89          iLeft := UDINT#0;
90          frac_tmp := PV / LREAL#0.139;
91          iLeft_0 := UDINT#0;
92          rtb_Product1 := frac_tmp;
93      ELSE
94          IF PV < LREAL#0.6 THEN
95              (* Binary Search *)
96              bpIdx := UDINT#3;
97              iLeft := UDINT#0;
98              iRight := UDINT#7;
99              WHILE (iRight - iLeft) > UDINT#1 DO
100                  IF PV < LookupTablenD2_bp01Data[bpIdx] THEN
101                      iRight := bpIdx;
102                  ELSE
103                      iLeft := bpIdx;
104                  END_IF;
105                  temp2 := UDINT_TO_DWORD(iRight + iLeft);
106                  bpIdx := DWORD_TO_UDINT(SHR(temp2, ...
107                      DINT_TO_USINT(UDINT#1)));
108              END_WHILE;
109              frac_tmp := (PV - LookupTablenD2_bp01Data[iLeft]) / ...
110                  (LookupTablenD2_bp01Data[iLeft + UDINT#1] - ...
111                  LookupTablenD2_bp01Data[iLeft]);
112          ELSE
113              iLeft := UDINT#6;
114              frac_tmp := (PV - LREAL#0.505) / LREAL#0.09499999999999973;
115          END_IF;
116          IF PV < LREAL#0.6 THEN
117              (* Binary Search *)
118              bpIdx := UDINT#3;
119              iLeft_0 := UDINT#0;
120              iRight := UDINT#7;
121              WHILE (iRight - iLeft_0) > UDINT#1 DO
122                  IF PV < LookupTablenD3_bp01Data[bpIdx] THEN
123                      iRight := bpIdx;
124                  ELSE
125                      iLeft_0 := bpIdx;
126                  END_IF;
127                  temp1 := UDINT_TO_DWORD(iRight + iLeft_0);
128                  bpIdx := DWORD_TO_UDINT(SHR(temp1, ...
129                      DINT_TO_USINT(UDINT#1)));
130              END_WHILE;
131              rtb_Product1 := (PV - LookupTablenD3_bp01Data[iLeft_0]) ...
132                  / (LookupTablenD3_bp01Data[iLeft_0 + UDINT#1] - ...
133                  LookupTablenD3_bp01Data[iLeft_0]);
134          ELSE
135              iLeft_0 := UDINT#6;
136              rtb_Product1 := (PV - LREAL#0.505) / ...
137                  LREAL#0.09499999999999973;
138          END_IF;
139      END_IF;
140      (* Sum: '<S1>/Sum1' *)

```

H.6 Kode for plott av resultat

```
131      (* Column-major Interpolation 1-D
132      Interpolation method: 'Linear point-slope'
133      Use last breakpoint for index at or above upper limit: 'off'
134      Overflow mode: 'wrapping' *)
135      (* Column-major Lookup 1-D
136      Search method: 'binary'
137      Use previous index: 'off'
138      Interpolation method: 'Linear point-slope'
139      Extrapolation method: 'Linear'
140      Use last breakpoint for index at or above upper limit: 'off'
141      Remove protection against out-of-range input in generated ...
142      code: 'off' *)
143      (* Prelookup - Index and Fraction
144      Index Search method: 'binary'
145      Extrapolation method: 'Linear'
146      Use previous index: 'off'
147      Use last breakpoint for index at or above upper limit: 'off'
148      Remove protection against out-of-range input in generated ...
149      code: 'off' *)
150      (* Column-major Interpolation 1-D
151      Interpolation method: 'Linear point-slope'
152      Use last breakpoint for index at or above upper limit: 'off'
153      Overflow mode: 'wrapping' *)
154      MV := SP - PV;
155      (* Product: '<S1>/Product1' incorporates:
156      * Lookup_n-D: '<S1>/Lookup Table (n-D)2'
157      * Lookup_n-D: '<S1>/Lookup Table (n-D)3' *)
158      rtb_Product1 := (((LookupTablenD3_tableData[iLeft_0 + UDINT#1] - ...
159      LookupTablenD3_tableData[iLeft_0]) * rtb_Product1) + ...
160      LookupTablenD3_tableData[iLeft_0]) * MV;
161      (* Sum: '<S1>/Sum' incorporates:
162      * DiscreteIntegrator: '<S1>/Discrete-Time Integrator'
163      * Lookup_n-D: '<S1>/Lookup Table (n-D)2'
164      * Product: '<S1>/Product' *)
165      MV := (((LookupTablenD2_tableData[iLeft + UDINT#1] - ...
166      LookupTablenD2_tableData[iLeft]) * frac_tmp) + ...
167      LookupTablenD2_tableData[iLeft]) * MV) + ...
168      c_DiscreteTimeIntegrator_DS;
169      (* Saturate: '<S1>/Saturation' *)
170      IF MV > LREAL#1.0 THEN
171      (* Output: '<Root>/MV' *)
172      MV := LREAL#1.0;
173      ELSIF MV < LREAL#0.0 THEN
174      (* Output: '<Root>/MV' *)
175      MV := LREAL#0.0;
176      END_IF;
177      (* End of Saturate: '<S1>/Saturation' *)
178
179      (* Update for DiscreteIntegrator: '<S1>/Discrete-Time ...
180      Integrator' *)
181      c_DiscreteTimeIntegrator_DS := c_DiscreteTimeIntegrator_DS + ...
182      rtb_Product1;
183      IF c_DiscreteTimeIntegrator_DS ≥ LREAL#1.0 THEN
184      c_DiscreteTimeIntegrator_DS := LREAL#1.0;
185      ELSIF c_DiscreteTimeIntegrator_DS ≤ LREAL#-1.0 THEN
186      c_DiscreteTimeIntegrator_DS := LREAL#-1.0;
187      END_IF;
188      (* End of Update for DiscreteIntegrator: '<S1>/Discrete-Time ...
189      Integrator' *)
190      (* End of Outputs for SubSystem: '<Root>/Subsystem' *)
191      END_CASE;
192      END_FUNCTION_BLOCK
```

H.6 Kode for plott av resultat

H.6.1 Kode for pumperegulator plott

H.6 Kode for plott av resultat

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % PI Pumpe P01
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 Tank_data1 = importdata("MATLAB\bachelor\NivÅ Tank 1.csv");
5 virkelig_hoyde1_PLS = Tank_data1.data(55132:55492, 1);
6 virkelig_paadrag1_PLS = Tank_data1.data(55132:55492, 4);
7 Tid1 = (0:1:360);
8 virkelig_hoyde2_SIM = Tank_data1.data(53914:54274, 1);
9 virkelig_paadrag2_SIM = Tank_data1.data(53914:54274, 4);
10 sim_hoyde = out.sim_P01{1}.Values.Data;
11 hoyde_ref = out.sim_P01{3}.Values.Data;
12 sim_tid1 = out.sim_P01{1}.Values.Time;
13 sim_paadrag_P01 = out.sim_P01{2}.Values.Data;
14
15 subplot(2,1,1)
16 hold on
17 plot(sim_tid1, sim_paadrag_P01*100,'b')
18 plot(Tid1, virkelig_paadrag1_PLS,'m')
19 plot(Tid1, virkelig_paadrag2_SIM,'g')
20 title('PÅdrag til pumpe P01')
21 xlabel('Tid [s]')
22 ylabel('PÅdra u_{P01}')
23 legend('Simulink PI', 'Sysmac PI', 'OverfÅrt PI')
24
25 subplot(2,1,2)
26 hold on
27 plot(sim_tid1, sim_hoyde*100,'b')
28 plot(Tid1, virkelig_hoyde1_PLS,'m')
29 plot(Tid1, virkelig_hoyde2_SIM,'g')
30 plot(sim_tid1, hoyde_ref*100,'r')
31 title('NivÅet i tank 1')
32 xlabel('Tid [s]')
33 ylabel('HÅyde [cm]')
34 legend('Simulink PI', 'Sysmac PI', 'OverfÅrt PI', 'Settpunkt')
```

H.6.2 Kode for ventilregulator plott

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % PI Ventil LV11
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 Tank_data = importdata("MATLAB\bachelor\PID_LV11.csv");
5 virkelig_hoyde = Tank_data.data(950:4800, 1);
6 virkelig_paadrag = Tank_data.data(950:4800, 3);
7 Tid = (0:1.5:5775);
8 sim_hoyde = out.sim_PID{1}.Values.Data;
9 hoyde_ref = out.sim_PID{3}.Values.Data;
10 sim_tid1 = out.sim_PID{1}.Values.Time;
11 sim_paadrag_LV11 = out.sim_PID{2}.Values.Data;
12
13 subplot(2,1,1)
14 hold on
15 plot(sim_tid1, sim_paadrag_LV11*100)
16 plot(Tid, virkelig_paadrag)
17 title('PÅdrag til ventil LV11')
18 xlabel('Tid [s]')
19 ylabel('PÅdra u_{LV11}')
20 legend('Simulert pÅdrag', 'Virkelig pÅdrag')
21
22 subplot(2,1,2)
23 hold on
24 plot(Tid, virkelig_hoyde)
25 plot(sim_tid1, sim_hoyde*100)
26 plot(sim_tid1, hoyde_ref*100)
27 title('PI respons')
28 xlabel('Tid [s]')
29 ylabel('HÅyde [cm]')
```

H.6 Kode for plott av resultat

```
30 legend('OverfÅ,rt PI', 'Simulink PI', 'Settpunkt')
```