



PROCES OG AUTOMATION 3

M7 EFTERÅR 2022

Obligatoriske Opgave 2

Forfattere:

Marius, Overgaard
studienr. 2019061

Malthe Mikjær, Christensen
studienr. 2019734

Bo Rajan Bach, Pedersen
studienr. 2021265

Magnus Voss, Overby
studienr. 2019704

Indhold

1	Indledning	1
2	Metode og teori	1
2.1	V-modellen	2
2.2	PID regulering	3
2.3	Kaskade-regulering	4
2.4	Programeringsformer	4
2.4.1	Ladder	4
2.4.2	FBD	4
2.5	HMI	4
3	Præsentation af Projektet	5
3.1	Kravspecifikation fra Kunden	5
3.2	Statisk og dynamisk analyse af processerne	7
3.3	Statisk analyse	7
3.4	Dynamisk analyse	9
3.5	Proces: Kondenstank	11
3.6	Proces: Fødevandskammer	15
3.7	Proces: Elektrodekammer	17
3.8	Proces: Damp til opsamlingstanken	18
3.9	HMI-display	23
4	Diskussion	25
4.1	Vurdering af Resultat	25
4.2	Vurdering af Metode	25
5	Konklusion	25
6	Bilag	26
6.1	Bilag: PID addon	26

1 Indledning

Der skal laves en ny styring til et eksisterende damp produktion anlæg. Der producerer damp ved hjælp af 2 højspændings elektroder. Der er 4 hoved processer som skal reguleres.

1. Kondenstank

I starten af processen skal kondenstanken fyldes. Hertil bruges to redundant pumper som reguleres med en drøvlede ventil. Denne ventil reguleres af PLC'en så der er et konstant niveau på 50% i kondenstanken.

2. Fødevandskammer

Fra kondenstanken pumpes vandet op i fødevandskammeret. Vandet ligger nu og cirkulere op i elektrodekammeret og det som ikke bliver til damp, løber ned igen. I fødevandskammeret skal vandstanden være mellem 45% og 100%. På vej tilbage til fødevandskammeret måles vandets lednings evne. Hvis dette er for højt, bliver noget af vandet udskiftet ved at åbne en ventil i bunden samtidig med at noget ny pumpes i.

3. Elektrodekammer

En pumpe pumper vandet fra fødevandskammeret op til elektrodekammeret. Det er vandstanden her som bestemmer hvor meget "dampeffekt" der producerer. Dette reguleres med en PID regulator.

4. Damp opsamling

Dampen lukkes ud gennem en ventil til opsamlingstanken. Ventilen åbnes først ved 9 bar, og reguleres til at holde trykket på 9,5 bar.

Derudover er der to essentielle programmer, som skal udarbejdes, nemlig:

1. Styring af Vandkvalitet

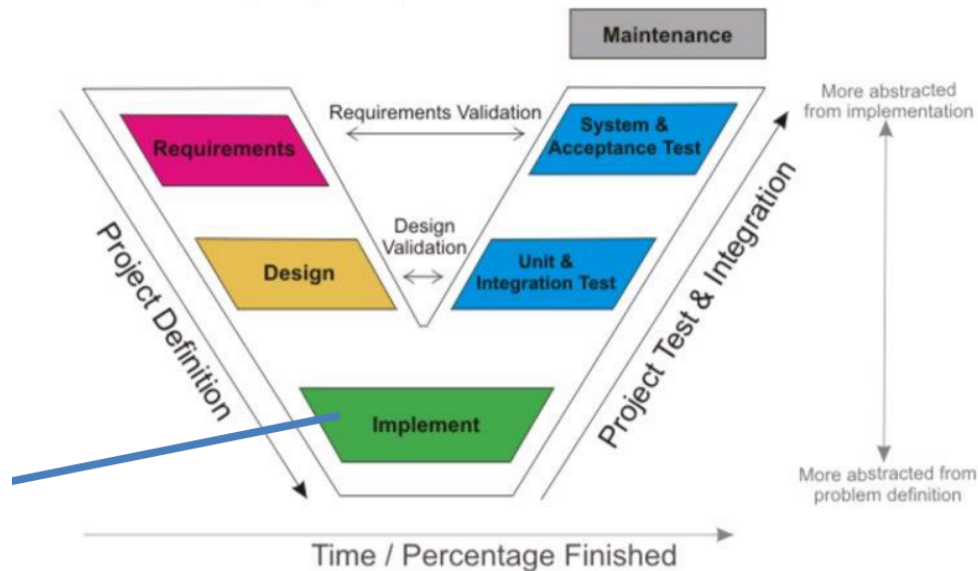
Vandkvaliteten styres ved at holde øje med ledningsevnen. Det gøres med en ledningstransmitter (CT-5). Vandet skiftes ud med en magnetventil MV-5).

2. HMI-display skal udarbejdes. Det skal vise processen og tillade start og stop. Det skal vise status på alle ventiler, niveauer, ledningsevne og tryk. Der skal også være mulighed for regulatorer og aflæse trends til niveau, tryk og åbningsgraden af V-105 og V-106. Dato og tid skal også vises.

2 Metode og teori

Metoden til udarbejdelse af projektløsning og rapport følger V-modellen. Delene til det udviklede software er blevet versionskontrolleret via github, hvor vi også har anvendt en todo-liste som backlock over opgaverne og et kanban-board til at sikre produktets fortsatte udvikling.

2.1 V-modellen

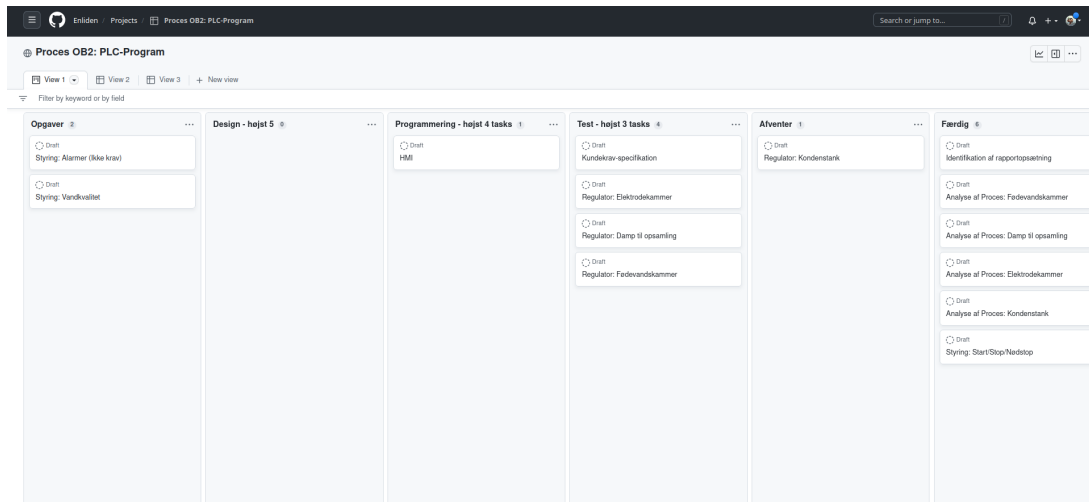


Figur 1: Skitsering af en simpel 5-trins V-model (hentet fra undervisnings-slides).

Delopgaverne (de 4 regulatorer, styring af vandkvalitet og programmering af HMI-displays) er uddeligeret til de fire gruppe-medlemmer med instruks om at udarbejde testskema til kravspecifikation (første trin: identifikation af kundekrav) og dernæst designe program-skitse til fælles fremlæggelse sammen med programteknisk test-ark (andet trin: Design af program og teknisk test). Efter godkendt fremlæggelse er selve udarbejdelsen af programkoden igangsat (tredje trin). Denne proces konvergerer ved sammensætning af programkoden, som derefter testes (fjerde trin: test af design). Hvis design-tests fejler, vender vi tilbage til design. Herefter testes kundekrav (femte og sidste trin). Hvis de fejler, vender vi tilbage til trin ét: identifikation og udspecificering af kundekrav.

Vi har valgt denne løsning i sammenhæng med et kanban board for nemt at kunne kontrollere status i udvikling og derfor have et overblik. Der indgår også en opdeling af opgaven i nemmere trin med en rækkefølge, så de fire gruppe-medlemmer kan overtage fra hinanden. Her spiller kanban's "done rules" særligt ind ved design, hvor designet skal fremlægges for de øvrige gruppe-medlemmer, før programmeringen påbegyndes. Modellen er også valgt pga. den cykliske effekt ved at teste programmet og vende tilbage til designfasen, hvis testen fejler.

Enkelte programdele (styring af vandkvalitet) fremgår ikke af opgaven, fordi de ikke er nået. Af samme grund er programdesign-tests desværre ikke blevet udførligt dokumenteret. Det fremgår desværre også af vores kanban:



Figur 2: Et billede af Kanban udført i Github "Projects".

2.2 PID regulering

PID regulering bygger på tre led der ud fra et fejl signal kan udregne hvad udgangssignalet skal sættes til fx en ventil. Disse er:

- P: Proportionalled. Dette er en forstærkning
- I: Integrableled. Dette fjerner den stående fejl der er ved ren P regulering, ved at integrere fejlen.
- D: Differentialeled. Dette regere på hældningen i fejlen og kan derved reagere hurtigt på ændringer.

Vi har i projektet brugt to forskellige PID-regulatorer

Magnus' egen PID regulator: Denne er bygget op efter formlen. (regulatoren kan ses under trykregulering)

$$U_k = K_p \cdot e + K_i \cdot \int e \cdot dt + K_d \cdot \frac{de}{dt}$$

Dette er i det kontinuerlige domæne, men studio5000 (og computere generelt) arbejder i det diskrete domæne, hvor formlen kommer til at se sådan ud:

$$U_d = K_p \cdot e + K_i \cdot \sum e \cdot \Delta t + K_d \cdot \frac{\Delta e}{\Delta t}$$

Rockwell PIDE blok: Deres PID regulator bruger hastighedsligning, som i diskret domæne skrives:

$$C_v = K_p \cdot (e_n - e_{n-1}) + K_i \cdot e \cdot \Delta t + C_{v,n-1} + K_d \cdot \frac{e_{e,n+2} - 2e_{n+1} + e_n}{\Delta t}$$

Den største forskel mellem de to regulatorer er, at PIDE forstærker en forskel i proportional-ledet, mens PID forstærker det nuværende fejl. *Fordelen* ved positionsligningen over hastighedsligningen fremgår, hvis udgangssignalet skal reagere voldsomt på store fejl og mindre på små fejl. Ønsker man i stedet en mere ensartet ændring i udgangssignalet ved en given ændring i fejlen e , skal man anvende hastighedsligningen. Hastighedsligningens proportionalled reagerer lineært på ændringen i udgangssignalet dvs. ens, når fejlen går fra 0 til 10, som når den går fra 45 til 55.

2.3 Kaskade-regulering

Kaskaderegulering bruges, når man bruger to forskellige procesværdier til at regulere processen. Reguleringen består af to transmittere (naturligvis) og to regulatorer. Den ene procesværdi oversættes af første regulator til et udgangssignal, der bruges som sætpunkt i den anden regulator, som regulerer udfra den anden procesværdi. Den første regulator er ”den overordnede reguleringssløjfe” (i vores tilfælde vil det være niveau-målingen) og den anden regulator er ”den underordnede reguleringssløjfe” (i vores tilfælde vil det så være flow-målingen). Det er ønskeligt, at den underordnede indreguleres hurtigere end den overordnede, så den tager små og hurtigere ændringer og dermed udjævner signalet som den overordnede regulator skal regulere efter. Hvis den overordnede skal indreguleres til at være hurtig, bør man ikke bruge kaskaderegulering. Man begynder altid med at indstille den underordnede regulering til en hurtigt indstilling ud fra et fast sætpunkt. **Formålet** må enten være, at den underordnede skal opfange og fjerne uønskede og generende forstyrrelser i den overordnede regulering (f.eks. en ulineær komponent, som den kan linearisere).

2.4 Programmeringsformer

I projektet fremkommer to programmeringssprog i større eller mindre omfang i de rutiner, som de studerende har udformet.

2.4.1 Ladder

Ladder er blevet brugt til på overskuelig måde at styre processen og aktivere boolske værdier. Her var også tiltænkt styring af vandkvalitet.

2.4.2 FBD

Funktionsblok-diagrammer er blevet anvendt ved de to regulator-rutiner og til at sammenligne procesværdi og sætpunkt i forbindelse med at lukke processen ved langvarige temperaturafvigelse. Generelt er dette sprog foretrukket ved beregninger med værdier, der ikke er boolske. Eksempelvis ville det også være blevet anvendt på regulatoren til elektrodekammeret.

2.5 HMI

HMI-displayet er også udarbejdet ud fra V-modellen, da der i første omgang er blevet lagt fokus på kundekrav og dernæst skitsering af design for forskellige versioner. Herefter blev HMI udarbejdet grafisk, og det er løbende blevet udvidet, da de relevante tags er blevet tilgængelige i forbindelse med de øvrige programmer. HMI'et er primært blevet testet med hensyn til tekniske krav og mindre i forhold til kundekrav.

3 Præsentation af Projektet

3.1 Kravspecifikation fra Kunden

Start/stop/nødstop

Kundens krav	Tilstand	Godkendelses parameter	Godkendt Ja/Nej
Start	Stoppet	Ved tryk på start: 1. Vand til fødevandskammer. (45cm - 100cm) 2. Pumpe fra fødevandskammer til elektrodekammer. (50cm - 100)	
Start dampproduktion	I drift	Aktivering af højspænding.	
Stop	I drift	Ved tryk på stop skal elektroderne fra kobles højspænding.	
Nødstop ved 10 bar i kedel i mere end 20 s.	I drift	Efter 10 bar i 20 sek. Skal processen stoppes automatisk. Start alarm.	

Figur 3: Kundekrav til start og stop af anlægget

Kondenstank.

Kundens krav	Tilstand	Godkendelses parameter	Godkendt Ja/Nej
Vand niveau (Sp)	N/A	Konstant 50%	
Skift pumpe mellem KP-1, KP-11	N/A	Skift pumpe lokalt (HMI)	

Figur 4: Kundekrav til kondenstanken

Fødevandskammer

Kundens krav	Tilstand	Godkendelses parameter	Godkendt Ja/Nej
Lav niveau	I drift	Aldrig under 45 cm. I normal drift	
Høj niveau	I drift	Aldrig over 100 cm. I normal drift	
Start pumpe	I drift, ≤ 50 cm.	Starter pumpe	
Stop pumpe	I drift ≥ 95 cm.	Stop pumpe	
Skift pumpe mellem KP-2, KP-22	N/A	Skift pumpe lokalt (HMI)	

Figur 5: Kundekrav til fødevandskammer

Dampproduktion

Kundens krav	Tilstand	Godkendelses parameter	Godkendt Ja/Nej
Vand niveau i elektrodekammer	I drift	Reguleres efter ønsket dampproduktion mellem 500-1000kW, dette svare til <u>en vandstanden</u> mellem 50-100cm	

Figur 6: Kunde krav til Elektrodekammer

Damp til opsamlingstank

Kundens krav	Tilstand	Godkendelses parameter	Godkendt Ja/Nej
Lukket ventil ved stop	Stoppet	Lukket ventil V-3	
Åbning af V-1	I drift	Ved 9 bar åbner V-1	
9,5 bar i kedlen	I drift	Regulator holder trykket i kedel til 9,5 bar.	

Figur 7: Kunde krav til damp til opsamling

Kedlens vandkvalitet

Kundens krav	Tilstand	Godkendelses parameter	Godkendt Ja/Nej
Vandets ledningsevne	I drift	Må aldrig overstige $100\mu S/cm$	
Udskiftning af vand i fødevandskammer	I drift	Åben V-5 hver 60sek i 10sek	
For høj ledningsevne $> 100\mu S/cm$	I drift	Åben V-5 i 10sek	
For høj ledningsevne $> 100\mu S/cm$ over 10sek	I drift	Frakoble højspænding, og åben V-5 i 35 sek. Hvis ledningsevne ikke ændres - stop drift.	

Figur 8: Kunde krav til vandkvalitet

HMI

Kundens krav	Tilstand	Godkendelses parameter	Godkendt Ja/Nej
Lokalt Start	Stoppet	Start proces	
Lokalt stop	I drift	Stop proces	
Valg af display	N/A	Skift display	
Status på ventiler	N/A	Vis om ventilerne er åbne, lukket og åbningsgrad	
Sensor data	N/A	Niveau, ledningsevne og tryk skal vises.	
Regulatorparametre	N/A	Regulatorparametre skal kunne indstilles	
Trendkurver	N/A	Trendkurver over niveau og tryk	

Figur 9: Kunde krav til HMI

3.2 Statisk og dynamisk analyse af processerne

For de fire delprocesser laves både statisk og dynamisk analyse. Dette gøres for at få indsigt i hvordan processen vil reagere.

Den statiske analyse bruges til at bestemme processens egen forstærkning. Vi kan også se om processen opfører sig lineært eller ulineært. Den dynamiske analyse, viser hvordan processen (PV) forløber over tid ved en ændring af belastningen eller sætpunktet fx. åbning af en ventil. Ud fra denne kan processens dødtid og tidskonstant aflæses.

3.3 Statisk analyse

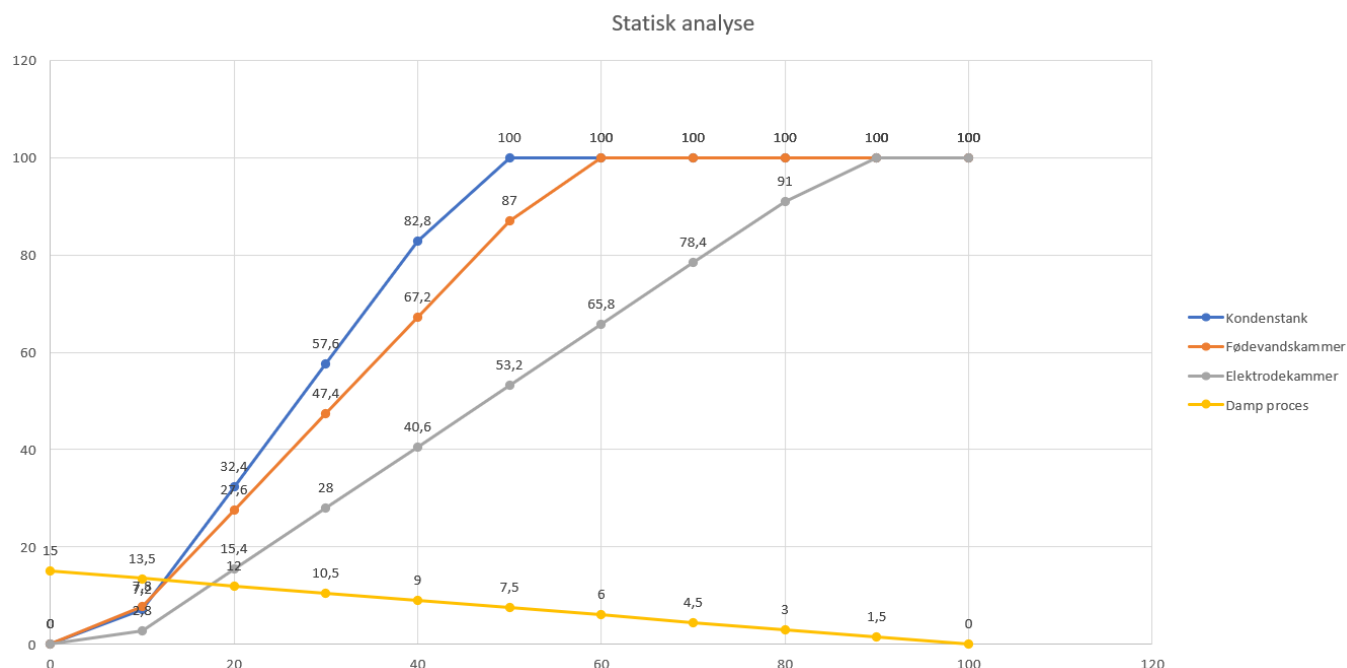
Process	Belastning	CV%										
		0	10	20	30	40	50	60	70	80	90	100
Kondenstank	15	0	7,2	32,4	57,6	82,8	100	100	100	100	100	100
Fødevandskammer	10	0	7,8	27,6	47,4	67,2	87	100	100	100	100	100
Elektrodekammer	7	0	2,8	15,4	28	40,6	53,2	65,8	78,4	91	100	100
Damp proces	N/A	15	13,5	12	10,5	9	7,5	6	4,5	3	1,5	0

Tabel 1: Statisk analyse af de fire delprocesser

Den statiske analyse bliver udført ved at skrue op for CV værdien for processen. I praksis ville det gøres ved fx at åbne en ventil. Vi undersøger, hvilket output PV vi får ved et given input CV.

Belastningen sættes til et passende størrelse. Dette gjorde vi ved at starte højt fx 40-50, for derefter at skrue ned til at vi får et output ved 10%.

Denne data, kan så sættes op som grafer, for at få et visuelt billede om lineariteten for processen.



Figur 10: Grafisk visualisation af statisk analyse. X-aksen er processen input-signal (CV), Y-akse er processens output (PV).

Kondenstank

Ud fra figur 10 kan det ses at mellem 10% til 40% er processen lineær. Vi kan også se, at hældningen er større end de andre, og derfor reagere kraftigere ved en ændring i CV. Man kan udregne processens egen forstærkning:

$$K_{p_{egen}} = \frac{\Delta CV_{40\%-30\%}}{\Delta PV_{40\%-30\%}} = \frac{82,8 - 57,6}{40 - 30} = 2,52$$

Fødevandskammer

Ud fra figur 10 kan det ses at mellem 10% til 50% er processen lineær. Vi kan også se, at hældningen er den anden største, og derfor reagere lidt svagere en kondenstank ved en ændring i CV. Man kan udregne processens egen forstærkning:

$$K_{p_{egen}} = \frac{\Delta CV_{40\%-30\%}}{\Delta PV_{40\%-30\%}} = \frac{67,2 - 47,4}{40 - 30} = 1,98$$

Elektrodekammer

Ud fra figur 10 kan det ses at mellem 10% til 80% er processen lineær. Vi kan også se, at hældningen er mindre end kondenstank og fødevandskammer. Den vil derfor reagere lidt svagere ved en ændring i CV. Man kan udregne processens egen forstærkning:

$$K_{p_{egen}} = \frac{\Delta CV_{40\%-30\%}}{\Delta PV_{40\%-30\%}} = \frac{40,6 - 28}{40 - 30} = 1,26$$

Damp proces

Ud fra figur 10 kan det ses at processen lineær over hele forløbet. Vi kan også se, at hældningen er negativ og mindre. Den vil derfor reagere svagest og faldene ved en ændring i CV. Man kan udregne processens egen forstærkning:

$$Kp_{egen} = \frac{\Delta CV_{40\%-30\%}}{\Delta PV_{40\%-30\%}} = \frac{9 - 10,5}{40 - 30} = -0,15$$

Vi kan så se, at da processen har en negativ forstærkning, betyder dette at den skal reguleres som en direkte proces.

3.4 Dynamisk analyse

Den dynamiske analyse bliver udført ved at importere de udgivende add-on instrukser for processerne. Belastningen holdes den samme som for statisk analyse. Her efter ændres CV fra 0 til 40%. Samtidig laves en trend over forløbet. Når outputet stabilt kan det nu analyseres. Der er to ting vi er interesseret i:

- **Dødtid i sekunder:**

Dødtiden er den tid før processen reagere på en input ændring. Ved en høj dødtid, kan processen være svær at regulere siden at der går lang tid inden regulatoren ser ændringen. For at modarbejde dette kan der bruges Feedforward.

- **Tidskonstant i sekunder**

Tidskonstanten (tau - τ) beskriver hvor hurtigt processen stiger. Tidskonstanten bliver defineret ud fra startværdien for PV og til den er noget 63% af den samlede stigning. Det kan udregnes for en indirekte proces som:

$$PV_{\tau} = ((|PV_{\text{efter}} - PV_{\text{før}}|) \cdot 0,63) \pm PV_{\text{før}}$$

Hvis det er en direkte proces skal der byttes rundt på nogle fortegn.

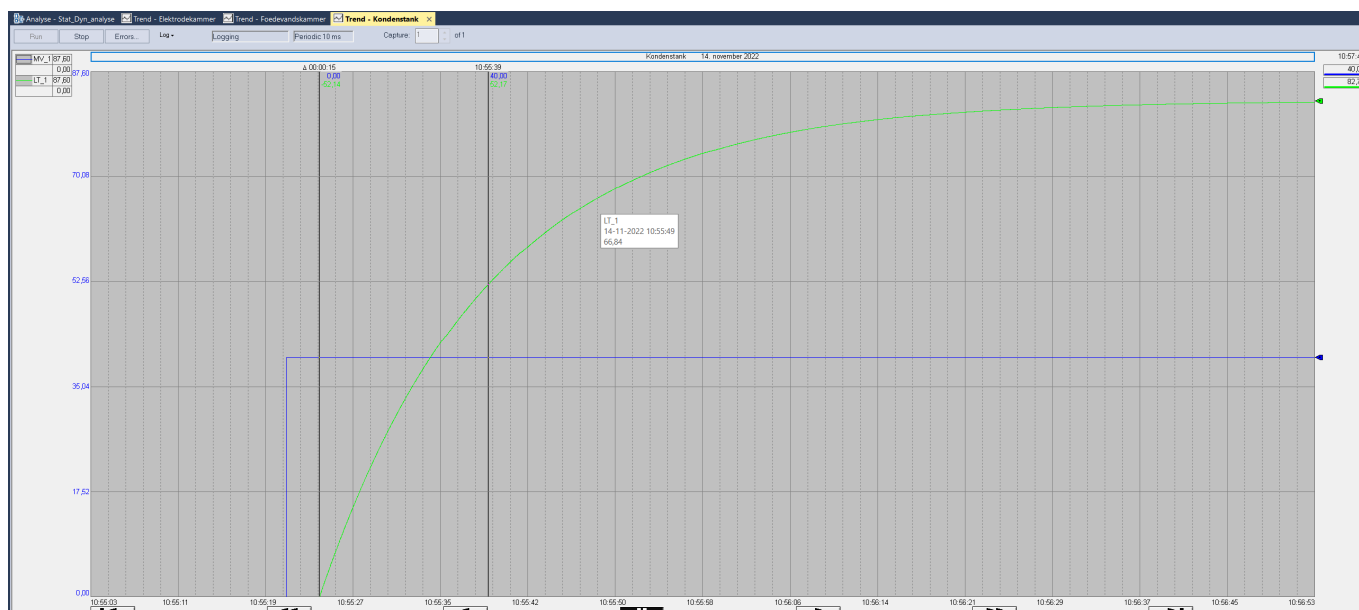
Tiden aflæses fra når der er en ændring i PV til PV_{τ} . Studie5000 gør dette nemt da man kan i en trend først sætter sin start value punkt. Derefter højre klikker og vælger Delta punkt. Denne sættes så ved $PV = PV_{\tau}$ og øverst kan konstanttiden så aflæses.

Hvis det ikke var et simuleret anlæg, ville man åbne en ventil, starte en pumpe, eller ligende. Derefter kan se hvordan anlægget reagere på denne step ændring. Man skal sørge for at man holder sig inde de tilladte værdier for anlægget.

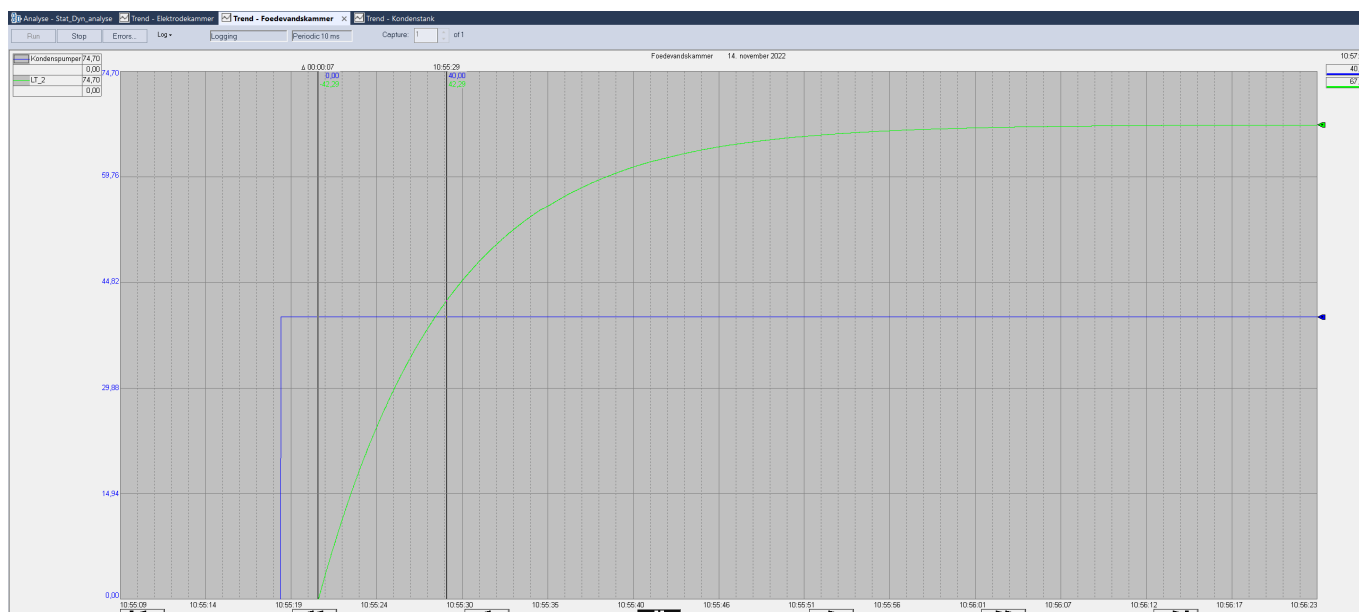
Ud fra de nedenstående trendkurver (figur: 11,12,13,14) findes dødtid og tidskonstant for de 4 processer.

	Dødtid [s]	Tidskonstant τ [s]
Kondenstank	1	6
Fødevandskammer	2	7
Elektrodekammer	2	15
Damp proces	1	10

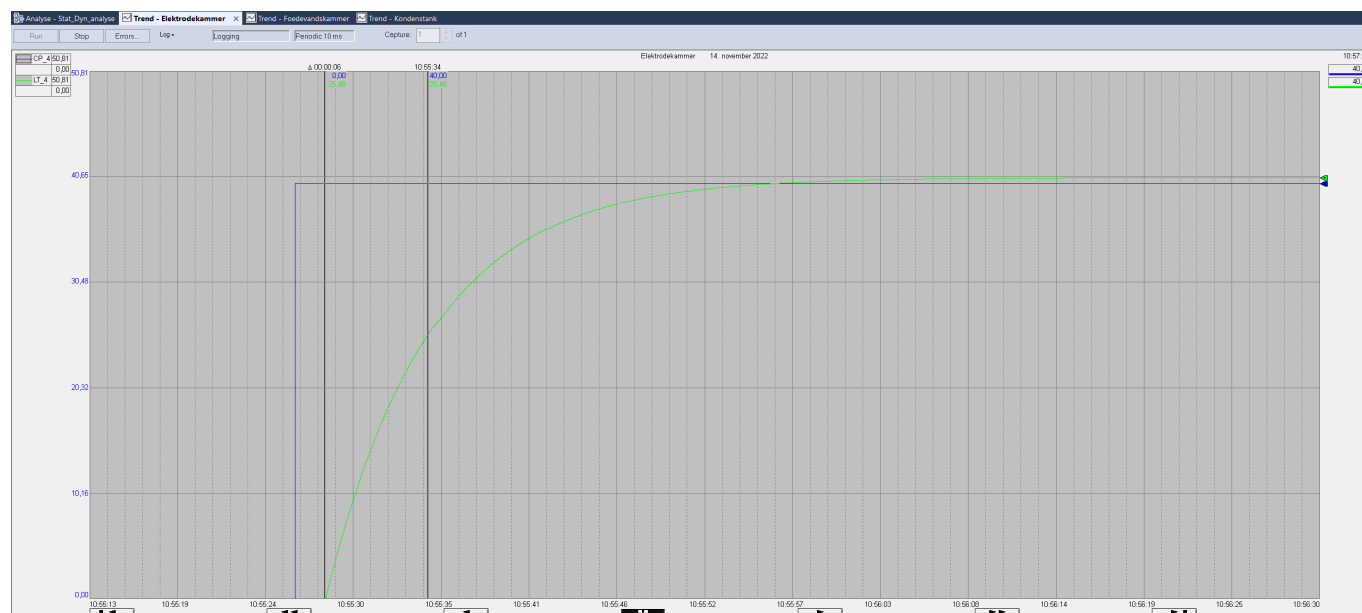
Tabel 2: Resultat fra dynamisk analyse, for de fire processer. Udarbejdet på baggrund af trendkurver.



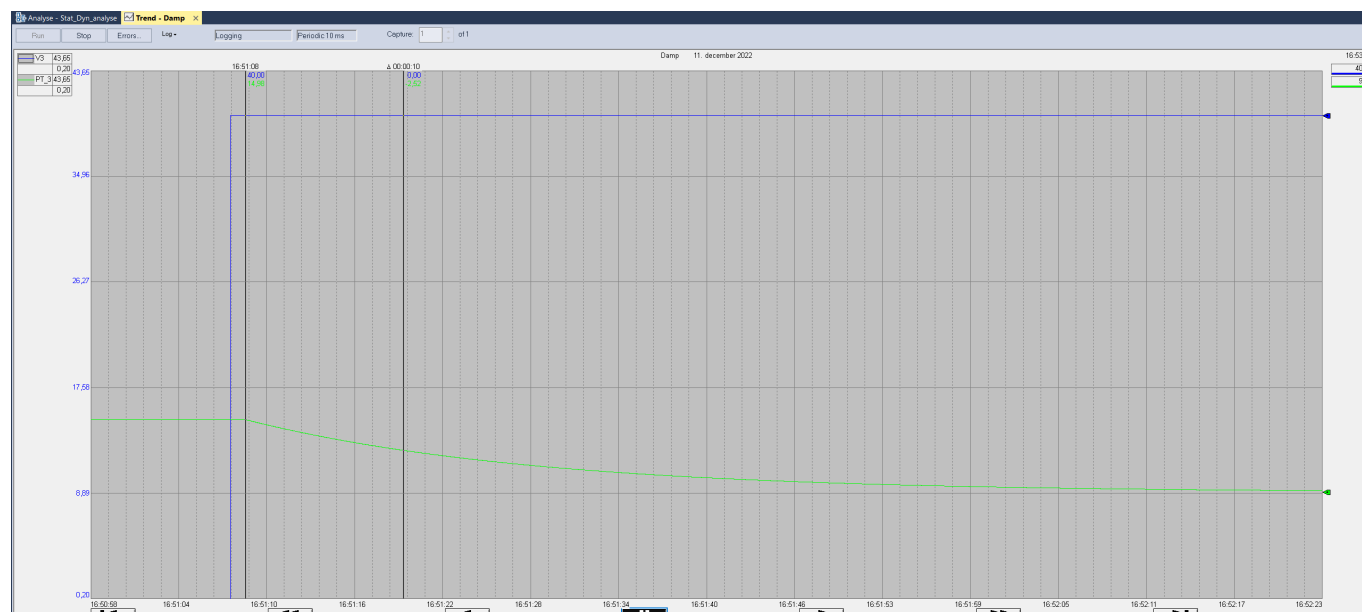
Figur 11: Dynamisk analyse af kondensstank processen ved 40%CV som step ændring



Figur 12: Dynamisk analyse af fødevandskammer processen ved 40%CV som step ændring



Figur 13: Dynamisk analyse af elektrodekammer processen ved 40%CV som step ændring



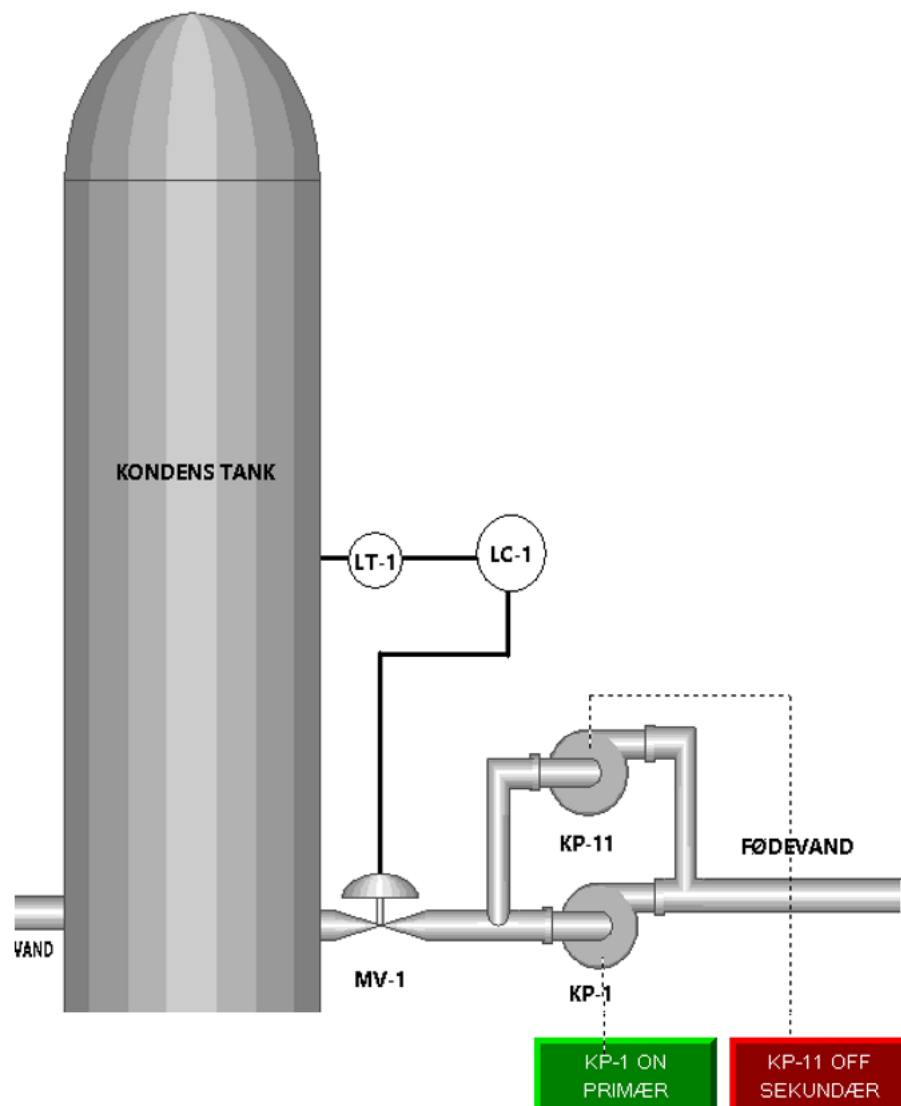
Figur 14: Dynamisk analyse af damp processen ved 40%CV som step ændring

3.5 Proces: Kondenstank

Krav: Kondens tanken skal påfyldes således at minimum 50% at tanken er fyldt med fødevand jv. kundes krav specifikationer. det skal foregå ved manuelt at skifte mellem de to fødevandspumper, hhv. primær pumpe (kP-1) og sekundær pumpe (KP-11). tilvalgsstyring af de to pumpe skal foregår gennem det integreret HMI display. Justering af påfyldning til tanken sker gennem regulering af magnetventil (MV-1) som er styret af ON/OFF signal.

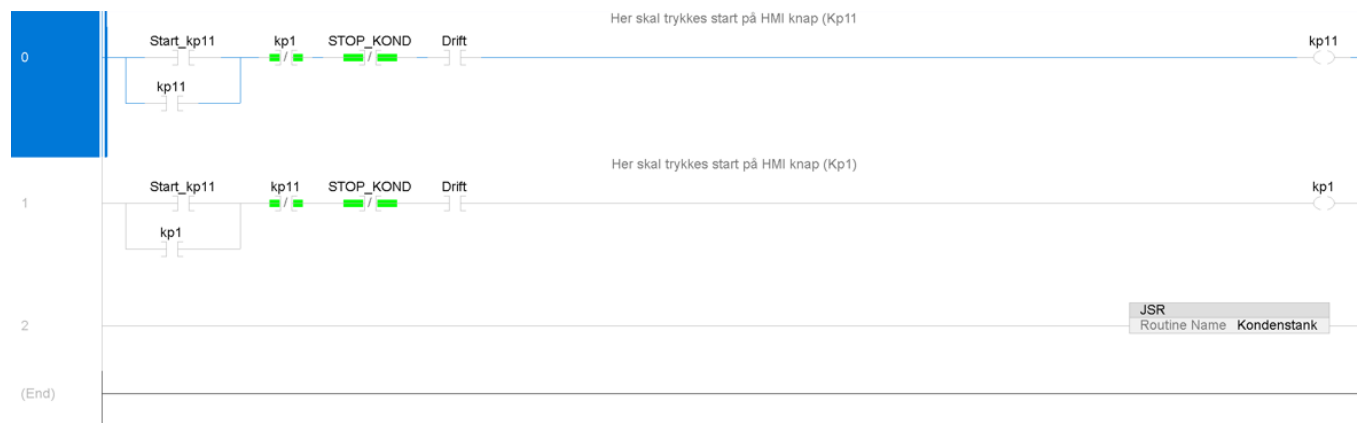
Kundekrav	Tilstand	Godkendelse af parameter	Godkendt Ja/Nej
Neutral	Fødepumper stoppet		ja
Start	KP-1 eller KP-11 aktiveres	1. MV-1 åbningsgrad afhænger af vandmængde i kondenstanken 2. Fyldning af Kondenstank over 50%	Ja
Skift mellem KP-1 eller KP-11	Manuelt tilvalg	Skift mellem pumperne sker på HMI displayet.	Ja

Figur 15: kondenstanks krav



Figur 16: kondens tank via HMI Display

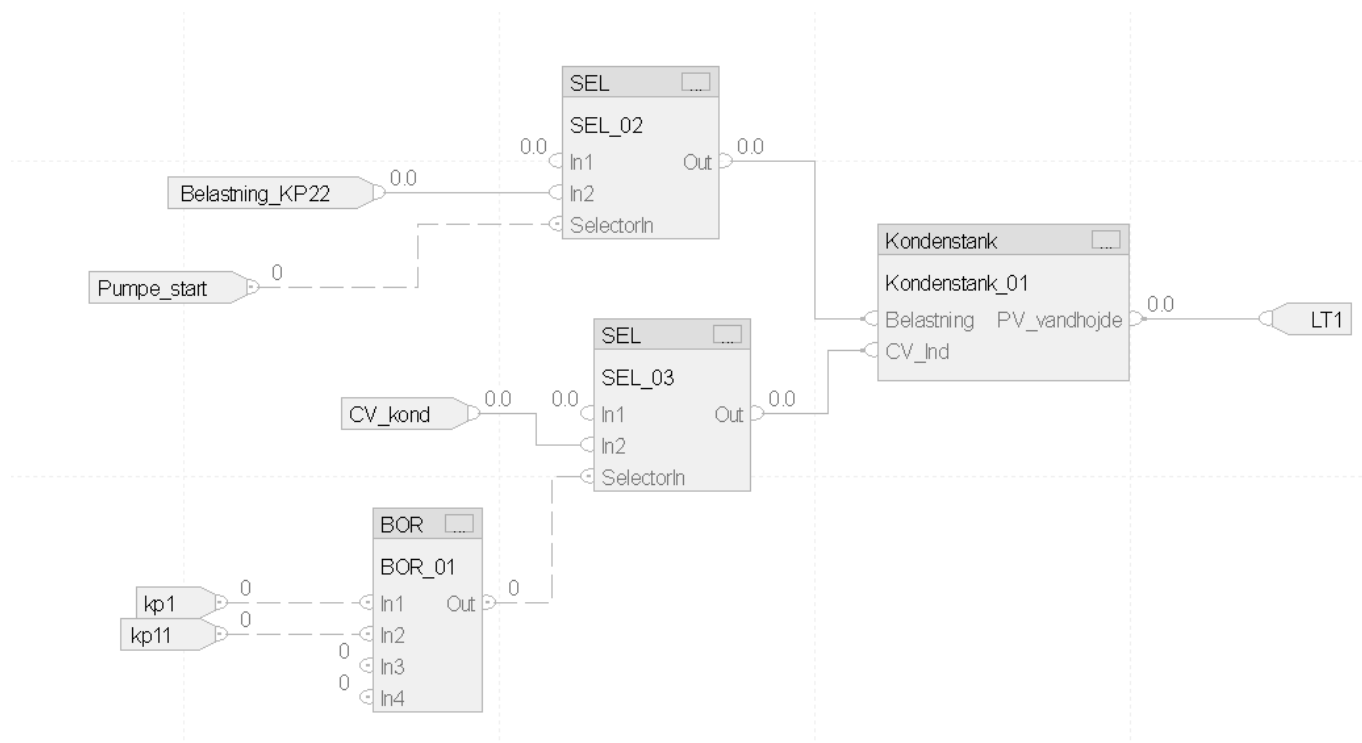
Ladder: Vi har valgt at benytte ladder programmering til On/Off styring af pumperne (KP-1 og (KP-11) dette er udarbejdet i forhold til kravspecifikationerne fra kunden om at kunne starte og stoppe fødepumperne manuelt.



Figur 17: kondenstank

Regulator og indregulering

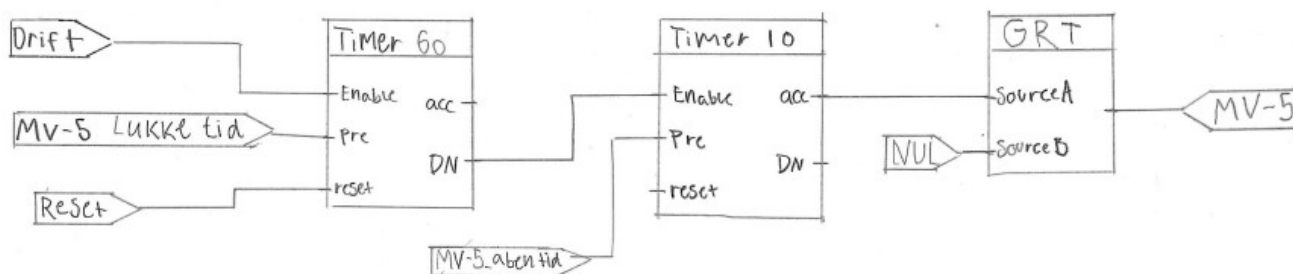
Vi har desværre ikke testet vores indregulering med grafer omkring kondenstanken, da tiden desværre har været begrænset. Vi har dog programmeret forholdene omkring kondens tanken. Vi har drøftet om det bedst kunne betale sig at lade påfyldningen af kondenstanken på omkring 50% eller om det bedre kunne svare sig rent vedligeholdelsemæssigt at lade den påfylde i et højere niveau for at undgå pumperne og MV-1 kørte oftere.



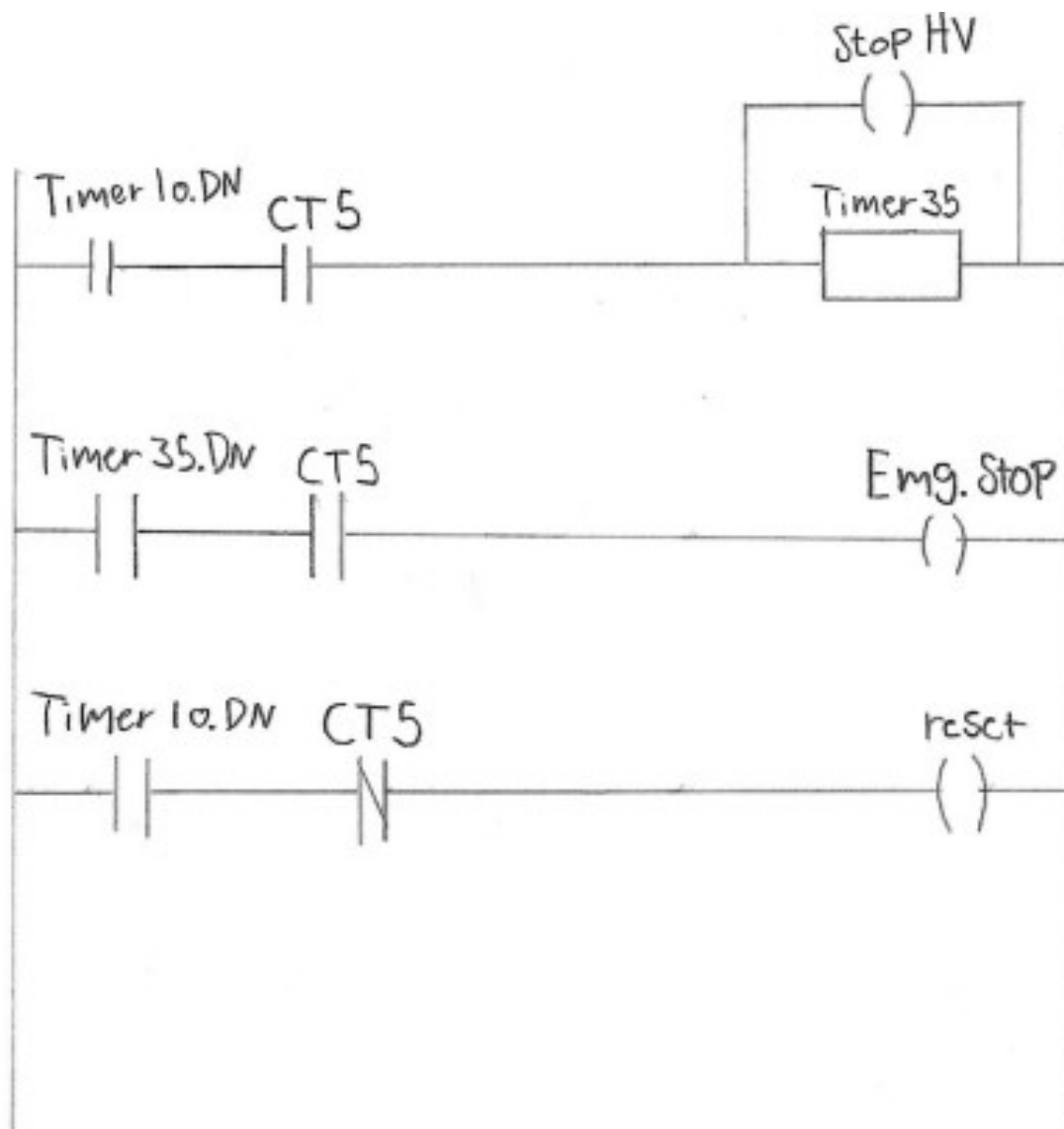
Figur 18: kondenstank

Teknisk krav	Tilstand	Godkendelses parameter	Godkendt Ja/Nej
Ved start skal der være den rigtige mængde vand i tanken.	I drift	For meget vand i tanken = Åben MV-5 For lidt vand i tanken = Start Pumper	Ja
Efter stop skal processen genstartes	I drift	Alle timere starter forfra når der bliver trykket på start	Ja
Nå MV-5 åbnes skal belastningen øges	I drift	Belastningsgraden skal stige nå MV-5 Åbnes	Ja

Figur 21: Tekniskekrav for fødevandskammer



Figur 22: Fødevandskammer timer



Figur 23: Design af Ladder programmeringen til åbning af MV-5 i tilfælde af ledningsevne problemer

Da denne del af anlægget er en ON/OFF regulering er der ikke nogen regulator at tage højde for, men der skal konstant være en vandhøjde mellem 45 og 95 cm. Dette opnås ved en LT måle vandstanden, og kan efter behov starte og slukke KP- 2 eller KP-22. Det er her kravet om vandets ledningsevne bliver styret, hvis ledningsevnen bliver for høj, vil ventilen MV-5 automatisk åbne i 35 sekunder. Hvis ledningsevnen herefter stadig er for høj vil den aktivere Emg. stop. Ventilen MV-5 skal automatisk åbne i 10 sekunder hvert minut. For at kunne opnå dette krav er der indsat tidsblokke der sørger for disse tider bliver overholdt.

3.7 Proces: Elektrodekammer

Note: Afsnittet er desværre udeladt pga. tidspres.

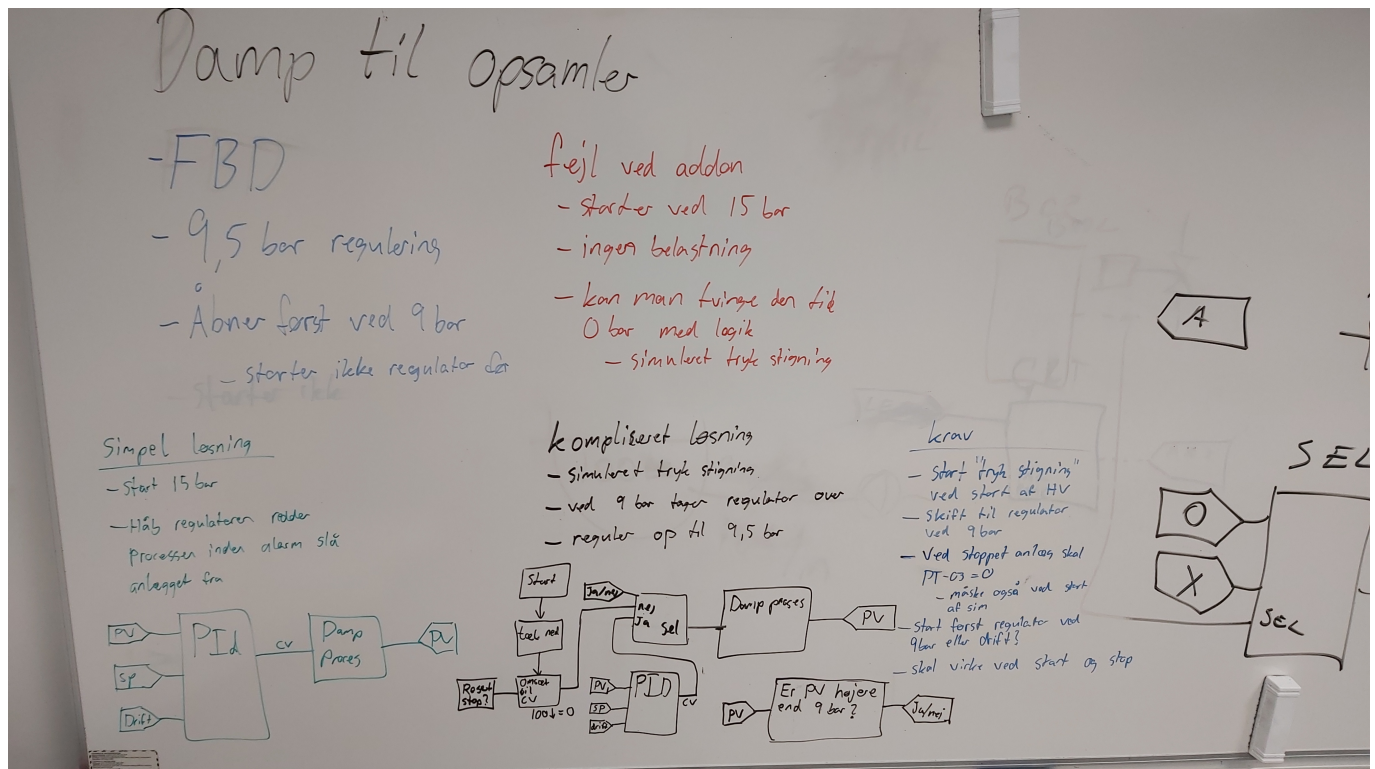
Regulator og indregulering

Note: Afsnittet er desværre udeladt pga. tidspres.

3.8 Proces: Damp til opsamlingstanken

Design og testkrav

Til dampprocessen blev der valgt at starte design fasen på tavlen. Her blev der først skrevet kravspecifikationerne op, som det skulle den overholde. Derefter blev der noteret nogle problematikker med den addon instruks som var blevet udleveret.



Figur 24: Design noter for damp opsamlings proces
Det

Problematikken med addon instruks var at den ville i normal drift ved opstart starte ved 15 bar, og det var så opgaven at få regulatoren til hurtigt nok at få den ned under 10 bar inden anlægget ville blive koblet fra. Vi gættede på at med lidt logik, ville det være muligt at lave et simuleret trykstigning. Derfor udarbejde vi to designs.

• Den simple løsning

Den simple løsning ville være at gøre som beskrevet før og regulere ned fra 15 bar. Dette følger bare ikke hvad opgaven for anlægget gik ud på, samt dette ville "fjerne" kravet og udfordringen om at den først skulle åbne ved 9 bar.

• Den komplicerede løsning

Hvis det var muligt at lave en simuleret trykstigning, ville man kunne lave en regulator som ville virke mere realistisk. Det var vigtigt ikke at ændre addon instruks da dette var udtrykkeligt sagt af underviseren.

Måden vi ville simulere et trykstigning, var at lave noget der talte ned fra 100 til 0 og sætte dette ind i CV for addon instruks. Dette ville simulere at vi startede ved helt åben ventil, og derved 0 bar,

og så stille lukkede ventilen og derved skabe en trykstigning. Ved 9 bar ville regulatoren så tage over ved brug af en select block.

Krav

Vi valgte at arbejde videre med den komplicerede løsning. Dette gjorde at vi bliver nødt til at sætte nogle tekniske krav, ikke kun til regulatoren, men også den simulerede trykstigning. De krav vi stiller til programmet er:

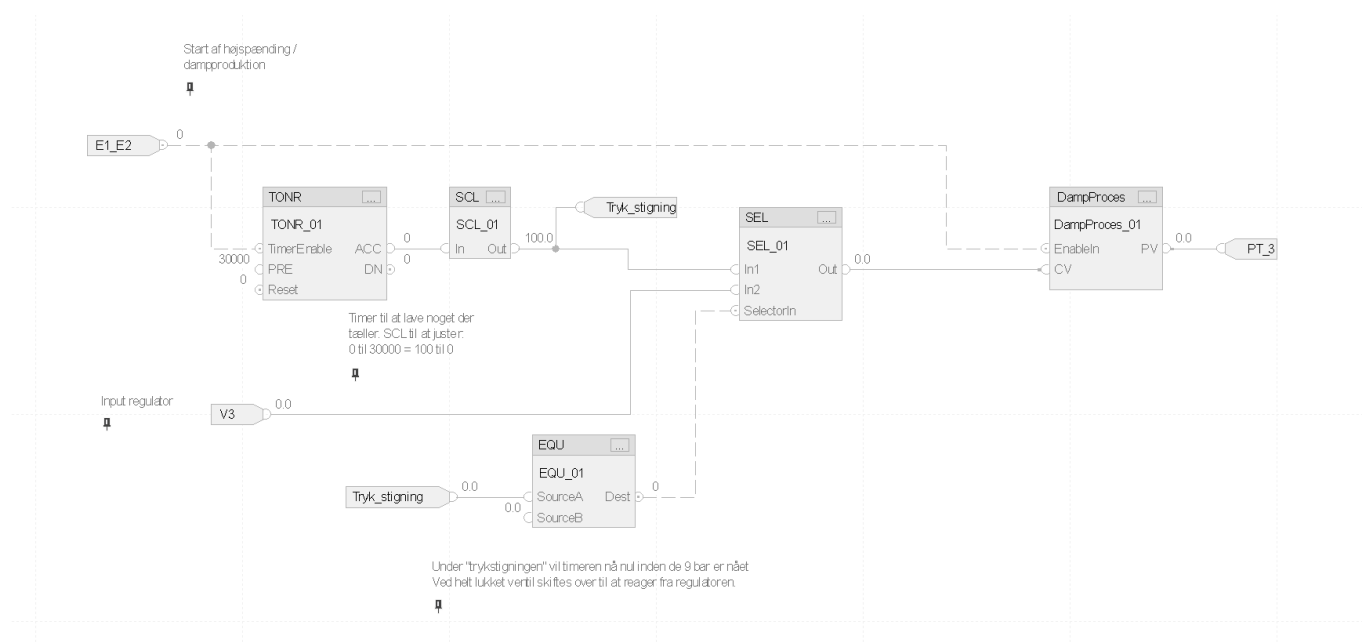
1. Start den simulerede trykstigning ved start af højspænding.
2. Skift til regulatoren ved 9 bar.
3. Ved stoppet drift og start af sim skal trykket falde til 0 bar.
4. Start først regulator ved 9 bar eller drift. (ved design fasen var det svært at bestemme hvad der ville give det bedste resultat.)
5. Skal virke ved start og stop

Ud fra disse krav kan vi nu starte på selve programmering i Studio5000.

Beskrivelse af programmet

Der skal bruges en regulator. Derfor vælges en PID-regulator som vi selv har lavet. Denne er kortet ned til en addon instruks, som er nem at implementere. Programmet for PID regulatoren kan ses i bilag 6.1. Først vises den endelige kode, efterfølgende bliver den beskrevet.

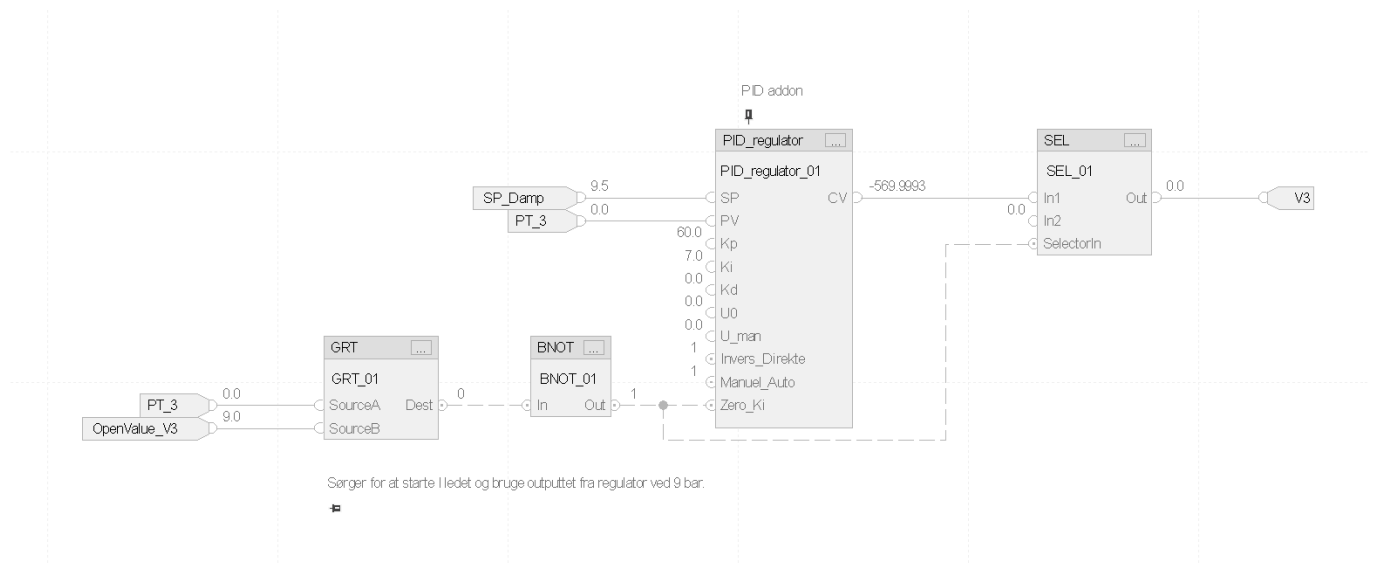
Trykstigning



Figur 25: Program code for simulering af trykstigning

Den simulerede trykstigning starter først ved start af højspænding (E1_E2). Derefter vil timeren (TONR_01) begynde at tælle op til 30000 (30 sekunder), som gennem en scale block (SCL_01) gør den tæller fra 100 ned til 0. Når den når 0 vel select blocken (SEL_01) skifte over til at bruge inputtet fra regulatoren. Da DampProces addon starter med et input på 100 ved start af simuleringen, starter den ved 0 bar.

Regulatoren



Figur 26: Regulator programmering for opsamling af damp

Programering er valgt at lave i FBD da dette gør det nemt at sammensætte funktioner. Der er brugt egen PID regulator frem for Rockwell PIDE regulator, da det her er nemmere at se hvad der sker under hjelmen. PID regulatoren vil stå og prøve at regulere lige så snart PLC'en er tændt. Det er først når PT_3 måler højere end 9 bar vil GRT_01 give signal til at I ledet i PID regulatoren sættes igang, samt at outputtet CV fra PID'en bliver brugt grundet SEL_01.

Formlen til at beregne CV i PID regulatoren er:

$$CV_k = K_p \cdot e + K_i \cdot \int e \cdot dt + K_d \cdot \frac{de}{dt}$$

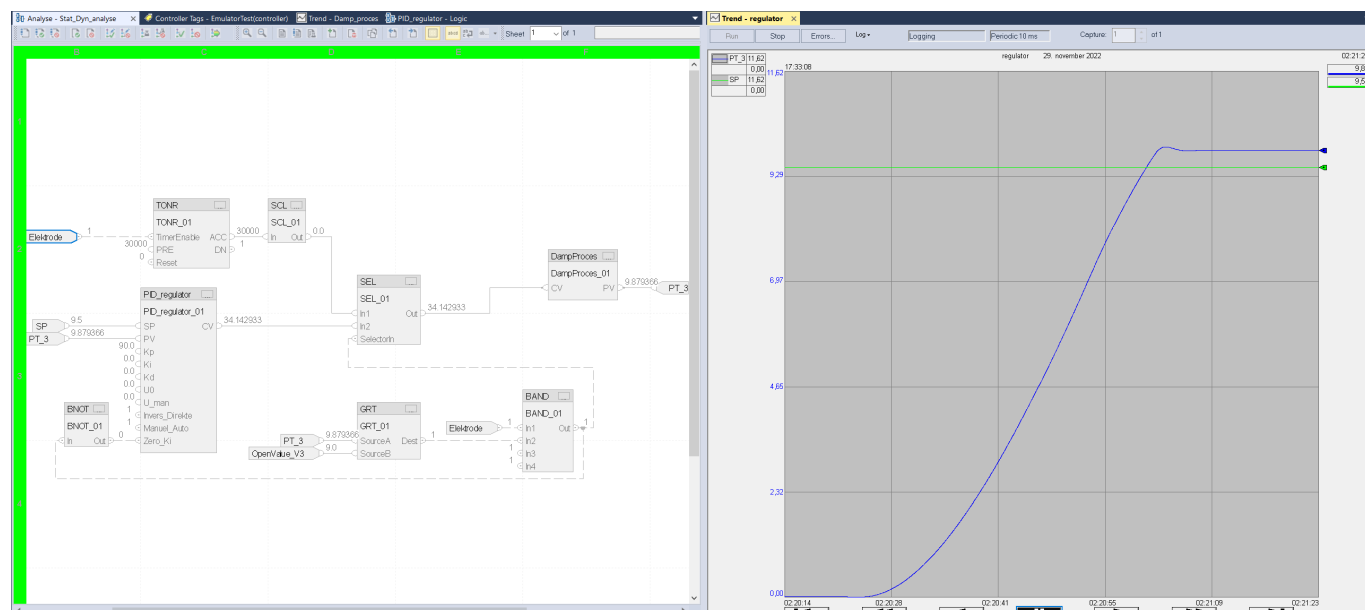
Dette er i det kontinuerlige domæne, men PLC'en arbejder i det diskrete domæne, derfor kommer formelen til at se sådanne ud.

$$CV_d = K_p \cdot e + K_i \cdot \sum e \cdot \Delta t + K_d \cdot \frac{\Delta e}{\Delta t}$$

Indregulering

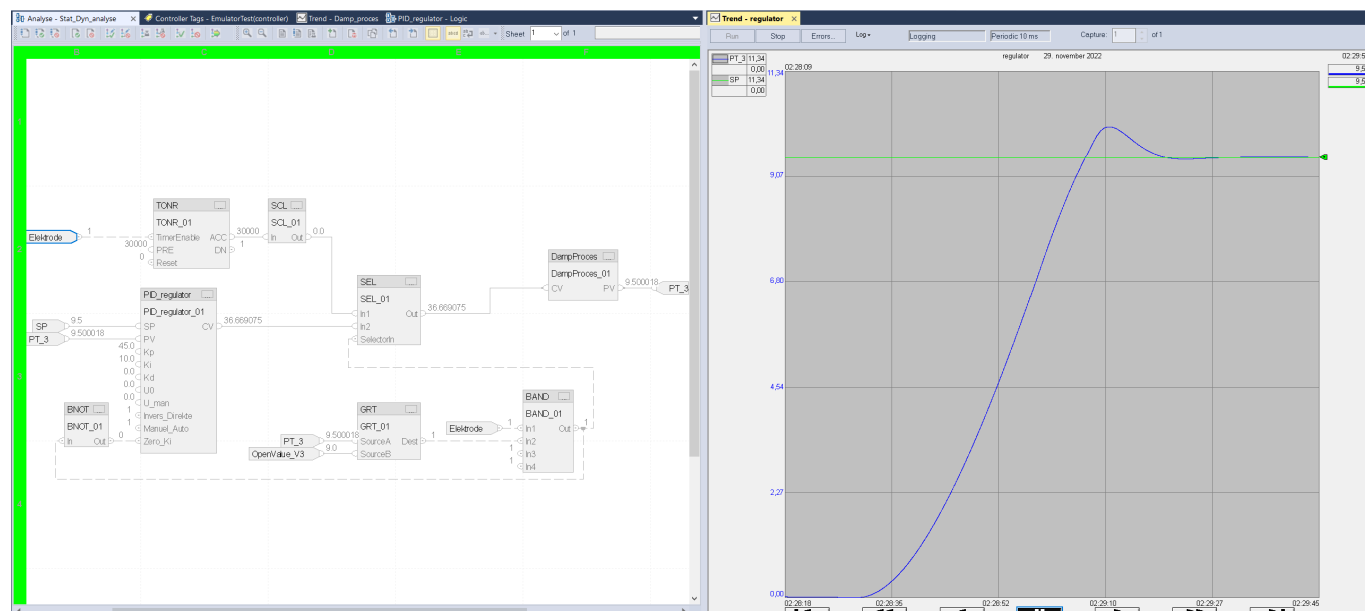
Til indregulering af damp processen bruges forsøgs metoden. Dette gøres da når der arbejdes med tryk, skal man være forsigtig at man ikke for lavet en bombe i stedet for. Derfor arbejdes der forsigtig og med små ændringer af gangen. Målet for regulering er et så lille over sving som muligt så vi ikke kkommer for længe over de 10 bar. Mellem hver ændring stoppes anlægget så trykket falder lidt inden der bliver prøves igen. Der bliv kigget på trendkurver for at holde øje med hvordan regulatoren arbejder.

Det første der gøres er at skrue stille op for Kp til at det bliver lidt uroligt. Vi kiggede efter at når der kom et lille undersving som skete ved Kp = 90. Efter dette, tages den halve værdi for Kp = 45.



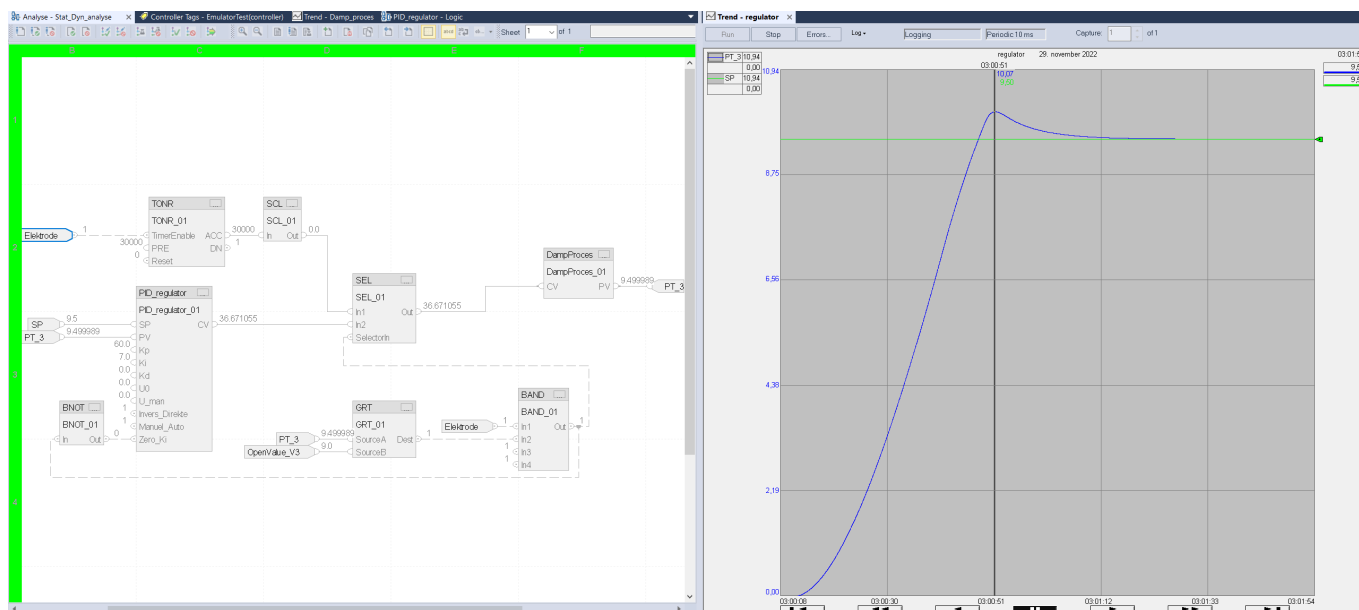
Figur 27: Indstilling K_p for damp processen. meget lille undersving efter oversvinge

Herefter indstilles K_i . Dette gøres efter at få en passende hurtigt fjernelse af den stående fejl, men uden at få et for stort oversving. $K_i = 10$.



Figur 28: Indstilling K_i for damp processen.

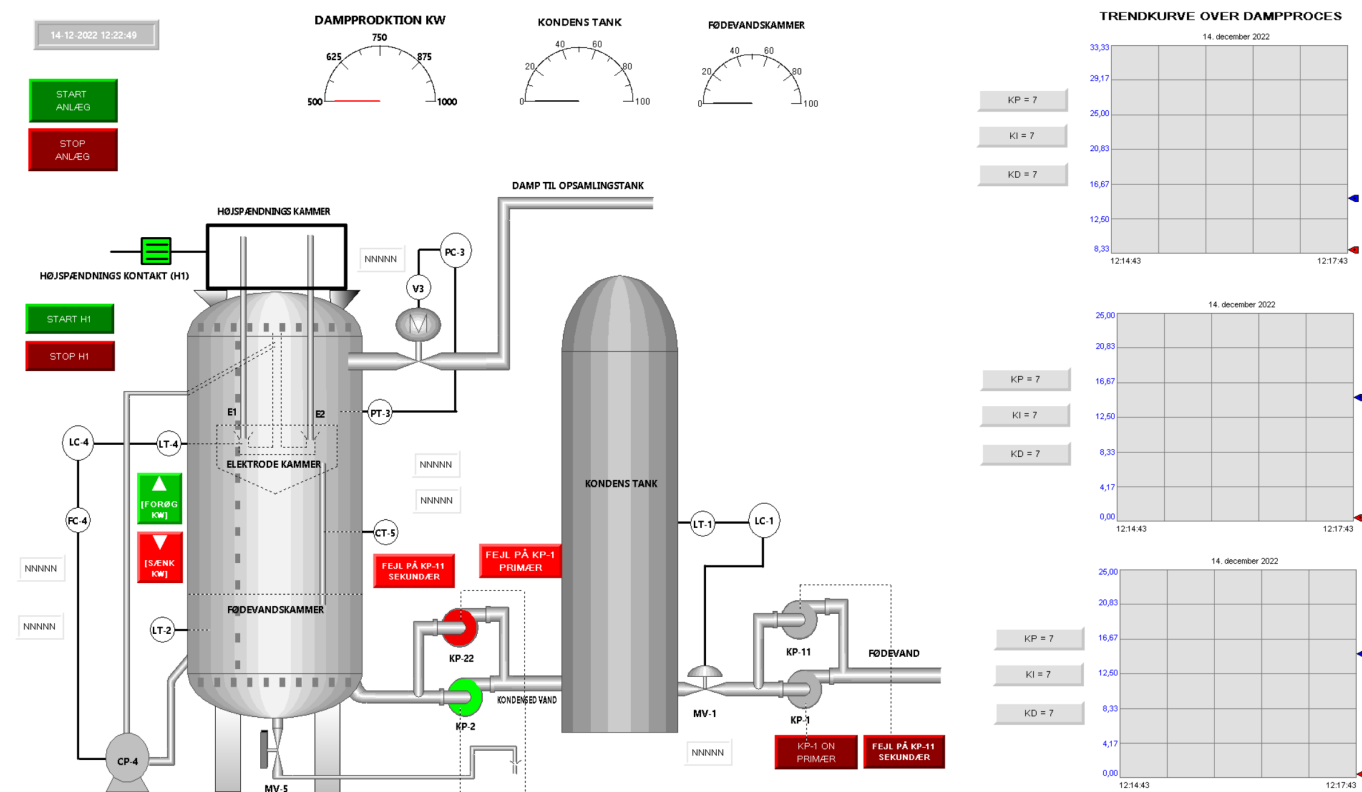
Til sidst fin justeres K_p og K_i til at få et så lille oversving som muligt. Vi endte ud med $K_p = 60$ og $K_i = 7$. Vi skruede lidt op for K_p for at få regulatoren til at reagere hurtigere, mens at vi skruede lidt ned for K_i da vi er lidt ligeglad med hvor hurtigt fejlen bliver fjernet, og dette gav os et lidt mindre oversving.



Figur 29: Fin indstilling K_p og K_i for damp processen. Målet var mindste oversving som muligt

Vi prøvede også med D ledet. Dette gjorde ikke rigtig noget positivt for vores mål, derfor udelod vi det. D ledet kan skabe problemer da den reagerer på hældningen af fejlen og derved kan reagere på støj hvis dette skulle være til stede.

3.9 HMI-display



Figur 30: HMI over processen

Vi har valgt at lave løsningen med HMI-display på én samlet skærm. Dette er gjort fordi proces på mange måder er simple i forhold til antal af ventiler mm, derfor ville det give det bedste overblik hvis kunden kun se helheden. Vi mangler desværre stadigvæk nogle enkelt forhold for indstillingsmuligheder for kunden, samt visninger.

Tag liste

Controller Tags - Program_Dampproduktion(controller) ✕ 🔗 Cross Reference

Scope: 📁 Program_Damppro ▼ Show: All Tags

Name	Value	Force Mask	Style	Data Type	Description	Constant
▶ AT_DP_kaska	{...}		{...}	PIDE_AUTOTUNE		<input type="checkbox"/>
Belastning	0.0		Float	REAL		<input type="checkbox"/>
Belastning_KP22	0.0		Float	REAL		<input type="checkbox"/>
CP_4	0.0		Float	REAL		<input type="checkbox"/>
CV_kond	0.0		Float	REAL		<input type="checkbox"/>
Drift	0		Decimal	BOOL		<input type="checkbox"/>
E1_E2	0		Decimal	BOOL		<input type="checkbox"/>
Emg_stop	0		Decimal	BOOL		<input type="checkbox"/>
Kd	0.0		Float	REAL		<input type="checkbox"/>
Ki	0.0		Float	REAL		<input type="checkbox"/>
Kp	0.0		Float	REAL		<input type="checkbox"/>
KP_2	50.0		Float	REAL		<input type="checkbox"/>
KP_22	50.0		Float	REAL		<input type="checkbox"/>
KP_Start	0		Decimal	BOOL		<input type="checkbox"/>
KP_Stop	0		Decimal	BOOL		<input type="checkbox"/>
kp1	0		Decimal	BOOL		<input type="checkbox"/>
kp11	0		Decimal	BOOL		<input type="checkbox"/>
Ledningsevne	0		Decimal	BOOL		<input type="checkbox"/>
LT1	0.0		Float	REAL		<input type="checkbox"/>
▶ Max_vandhojde	95		Decimal	DINT		<input type="checkbox"/>
Min_vandhojde	50.0		Float	REAL		<input type="checkbox"/>
MV_5	0		Decimal	BOOL		<input type="checkbox"/>
▶ MV_5_lukke_tid	60000		Decimal	DINT		<input type="checkbox"/>
▶ MV_5_aaben_tid	10000		Decimal	DINT		<input type="checkbox"/>
Nul	0.0		Float	REAL		<input type="checkbox"/>
Nul_Last	0.0		Float	REAL		<input type="checkbox"/>
OpenValue_V3	9.0		Float	REAL		<input type="checkbox"/>
PT_3	0.0		Float	REAL		<input type="checkbox"/>
Pumpe	0.0		Float	REAL		<input type="checkbox"/>
Pumpe_start	0		Decimal	BOOL		<input type="checkbox"/>
Pumpe_valg	0		Decimal	BOOL		<input type="checkbox"/>
▶ PV	0		Decimal	DINT		<input type="checkbox"/>
PV_vandhojde	0.0		Float	REAL		<input type="checkbox"/>

Reset	0		Decimal	BOOL		<input type="checkbox"/>
SP	0.0		Float	REAL		<input type="checkbox"/>
SP_Damp	9.5		Float	REAL		<input type="checkbox"/>
Start	0		Decimal	BOOL		<input type="checkbox"/>
Start_HV	0		Decimal	BOOL		<input type="checkbox"/>
Start_kp11	0		Decimal	BOOL		<input type="checkbox"/>
Stop	0		Decimal	BOOL		<input type="checkbox"/>
Stop_HV	0		Decimal	BOOL		<input type="checkbox"/>
STOP_KOND	0		Decimal	BOOL		<input type="checkbox"/>
▶ SUB_01	{...}	{...}		FBD_MATH		<input type="checkbox"/>
▶ timer_35	{...}	{...}		TIMER		<input type="checkbox"/>
Tryk_stigning	0.0		Float	REAL		<input type="checkbox"/>
V3	0.0		Float	REAL		<input type="checkbox"/>
Ventil_last	10.0		Float	REAL		<input type="checkbox"/>

Figur 31: Tagliste fra Studio5000

4 Diskussion

4.1 Vurdering af Resultat

Med udgangspunkt i den tid der er lagt i projektet, har vi opnået størstedelen af kundekravene. Der er stadig små mangler nogle steder og hele afsnit, der er udeladt, men der er vist, at gruppen kan udarbejde løsninger ved brug af PID-regulering samt udtænke sammensætninger af en simulator med add-on instruktioner.

4.2 Vurdering af Metode

Vi har arbejdet systematisk ved brug af V-modellen men er trods alt ikke kommet helt i mål med test af programmering, bl.a. fordi visse dele ikke er færdiggjort. Opdelingen af opgaver efter V-modellen trin 2-3-4 har muliggjort, at vi alligevel kunne komme i mål med størstedelen af opgaven, og det havde ikke været muligt på den måde uden at bruge V-modellen. Samtidig har V-modellen også skabt et overskueligt grundlag for en kanban, som har øget bevidstheden om, hvor langt vi er nået og kvaliteten af vores produkt.

5 Konklusion

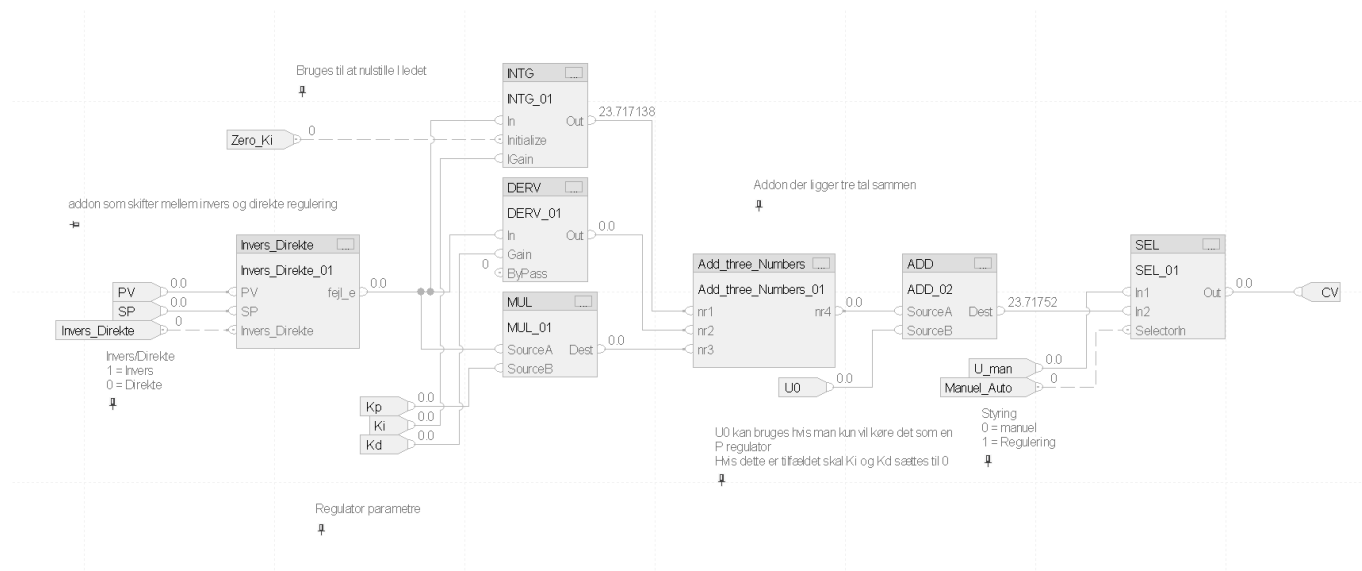
Vi har udarbejdet en kravspecifikation over de krav, der bliver stillet af opgaven. Der er systematisk arbejdet med design af løsning, programmering og sammensætning samt test af programmerne efter V-modellen. Der er desuden lavet statisk og dynamisk analyse over de fire delprocesser, hvor vi har observeret, at alle processer kan beskrives som 1.ordenssystemer.

Der er udarbejdet regulering af hver alle delprocesser, indregulering og test af regulatoren til elektrodekammeret er dog ikke blevet færdig. Ligeledes mangler der også en styring af vandkvaliteten. Projektets nuværende stadie fremgår af et kanban i Github Projects tilknyttet det 'repository', hvor programfilerne også ligger.

Til hele processen er der også udarbejdet et HMI-display, der også er udviklet systematisk med efter V-modellen og i kanban-systemet. Her ville processen kunne blive styret fra.

6 Bilag

6.1 Bilag: PID addon



Figur 32: PID regulator som addon instruks